CHAPTER III

THE FPGA IMPLEMENTATION OF PULSE WIDTH MODULATION

3.1 Introduction

A number of PWM schemes are used to obtain variable voltage and frequency supply. The Pulse width of PWM pulse varies with the Sine wave so as to restrain the lower order harmonics with easy control and large DC utilization. With successively improving reliability and performance of digital controllers, the digital control techniques have predominated over other analog controlled parts [45].

The proposed PWM generation unit is based on a specially designed, synchronous binary counter, resulting in maximum PWM frequencies with an adjustable duty cycle resolution, while the PWM unit can be easily interfaced to a microcontroller or DSP system. The development of high frequency PWM generator architecture for power converter control using FPGA [1]. The resulting PWM frequency depends on the target FPGA device speed grade and the duty cycle resolution requirements.

3.2 Comparator PWM Generator using FPGA

The Comparator PWM generator is one of the easiest methods of implementation of PWM. In this research comparison based upon the FPGA implementation of three types of PWM Generation. Based on the comparison results, the PWM Generation takes the least amount of area utilization in FPGA. The system input is an N-bit data word corresponding to the desired PWM duty cycle value, so that it can be easily interfaced to a micro-controller unit I/O port pins [17].
Fig. 3.1 Block Diagram of the Comparator PWM Generation

The system input is an N-bit register output, containing the N-bit data input, is compared with the output value of an N-bit free running synchronous counter by means of a comparator. When these two values become equal, the comparator output is used to reset the R/S latch output which produces the PWM wave [2].

The R/S latch output is set when the counter reaches an overflow condition at the end of a PWM period. Also the counter overflows signal is used to load the N-bit data input to the input register. So that the PWM output duty cycle change is performed at the
new PWM wave period in order to avoid any frequency and/or phase jitter of the output waveform [46].

![Simulation result of generation of Comparator PWM using Modelsim.](image)
Fig. 3.3 Synthesis result of generation of Comparator PWM for area utilization.
Fig. 3.4 Synthesis result of generation of Comparator PWM for delay and Speed.
Fig. 3.5 Synthesis result of generation of Comparator PWM for power consumption.
Fig. 3.6 Synthesis result of generation of Comparator PWM for Hardware utilization.
3.3 Counter Based PWM Generator using FPGA

This PWM Generator architecture is used for low power switching supplies. The architecture shown in fig. 3.7 is based on the principle that due to triggering of a counter by clock signal, clock is set equal to some multiple of switching frequency with help of a counter. The PWM output signal is set high before the clock signal and it remains high until it is reset after the counter value becomes equal to the duty cycle value. A counter is triggered by a clock signal, fclk, with frequency equal to some multiple of the power converter switching frequency, fsw [6]. The PWM output signal is set to logical “1” at the beginning of the switching period and is reset after a number of clock cycles equal to the integer value of the N-bit data input. The disadvantage of this method is that a high clock frequency is required, resulting in increased power dissipation [49].

Fig. 3.7 Block diagram of the Counter Based PWM generator.
Fig. 3.8 Simulation result of generation of Counter PWM using Modelsim.
Fig. 3.9 Synthesis result of generation of Counter PWM for area utilization.
Fig. 3.10 Synthesis result of generation of Counter PWM for delay and frequency utilization.
Fig. 3.11 Synthesis result of generation of Counter PWM for power consumption.
Fig. 3.12 Synthesis result of generation of Counter PWM for hardware utilization.
3.4 Delay PWM Generators

The delay PWM is one of the most utilized methods of Pulse Width Modulation Technique by which it can be produced by using a simple logic circuit in order to get an output in a very simple manner, the delay PWM works based on the block diagram given in the below fig.3.13 [32].

![Fig. 3.13 Block diagram of the Delay PWM generator.](Image)

It consists of data register, up/down counter and t-Flip flop. The up/down counter decides the direction of flow of PWM in output, the input data is given in the register and in the up/down counter whenever the clock pulse is gets high the data is fetch to the toggle flip-flop, after reset pin gets high, get the PWM output [54]. In the delay-line-based PWM circuit, a pulse from a reference clock starts a cycle and after a certain delay, designed to match the propagation delay experienced through the multiplexer, sets the PWM output to high [64] and [65]. The reference pulse propagates through the delay line and when it reaches the output selected by the multiplexer its value is used to set the PWM output to low. The total delay through the delay line is calculated to be equal to the reference clock period [19].
Fig. 3.14 Simulation result of generation of Delay PWM using ModelSim.
Fig. 3.15 Synthesis result of generation of Delay PWM for area utilization.
Fig. 3.16 Synthesis result of generation of Delay PWM for delay and Frequency utilization.
Fig. 3.17 Synthesis result of generation of Delay PWM for power consumption.
Fig. 3.18 Synthesis result of generation of Delay PWM for hardware utilization.
3.5 Result and Discussions

Simulation is carried out using ModelSim 6.3c and Synthesis is carried out using Xilinx 10.1. Hardware Implementation is performed by FPGA Spartan3 XC3S50. All the methods are tested by simulation and results are analyzed from the Synthesis report. The results are shown in table 3.1.

Fig. 3.19 Architecture of FPGA Spartan3 XC3S50 device.
FPGA stands for Field Programmable Gate Array. That’s the chip, which is programmed by user to perform the desired functionality. There are many possibilities of programming (types of FPGA):

1. Antifuse technology - programmed only once
2. Flash based - programmable several times
3. SRAM based - programmable dynamically

The last possibility is dominating technology. It allows very fast, unlimited in system reprogramming. This type of FPGA chip is used for irradiation tests. Architecture of FPGA from different vendors may differ but the main idea is almost the same. There will be shown on the example of Xilinx Spartan-IIE chip. FPGA architecture consists of:

   Flexible, programmable Configurable Logic Blocks (CLB), which provide the functional elements for constructing logic. Programmable Input/output Blocks (IOB), which provide the interface between the package pins and the internal logic. Delay-Locked Loops (DLL) for clock distribution. Dedicated internal memory (Block RAM). Versatile multi-level interconnects structure [24].

   LBs form the central logic structure with access to all support and routings. IOBs are located around logic and memory elements for easy routing signals from and to chip. Values stored in static memory cells control all the configurable logic elements and interconnect resources. These values must be loaded into the memory cells on power-up and can be reloaded to change function of the device, almost any time. Logic Cell (LC). Two LCs form slice and two slices form CLB. Each LC includes a 4-input function generator, carry logic and a storage element. The function generator is implemented as 4-input look-up table (LUT).

   The IOB features inputs and outputs that support a wide variety of I/O signaling standards. It consists of:
1. Three registers, which can work either as edge-triggered D-type flip-flops or as level-sensitive latches.
2. Programmable delay element in the input path, which eliminates pad-to-pad hold time.
3. Two multiplexers to control output from FPGA.

**Delay-Locked Loop**
This block eliminates skew between the clock input pad and internal clock-input pins throughout the device. Additionally delay is introduced such that clock edges reach internal flip-flops exactly one clock period after they arrive at the input.

**Block RAM**
That’s internal memory for used to store the data inside the chip. The word length can be configured by user.

**DDR memory**
The FPGA will be programmed with DDR interface. It will perform also test of this memory and report all the errors to the PC computer. During experiment PC computer will perform FPGA memory configuration test and in case of SEU errors in configuration it will reprogram FPGA. DDR memory will be refilled when too many errors will occur.

**Other components on FPGA board**
Other components like: voltage regulators, RS232C interface, oscillators seem to be less sensitive to the radiation. These elements will be tested passively during reset of experiment. In case of any problems the board will be tested outside the tunnel to know what part is broken.

**Shields for electronic boards**
According to data collected in LEDs, optical fiber and TLSs experiments there will be designed using proper shielding for neutron and gamma radiation. All proposed shields
will be tested in accelerator tunnel with all tests mentioned before. The goal of this test is to design such shield for electronic board that gives the least number of errors.

**Configuration**

This is the process by which the bit stream of a design is loaded into the internal configuration memory of FPGA. In experiment, it’s done by JTAG port.

Comparator PWM, Counter PWM and Delay PWM is implemented in this Spartan3 XC3s50 device. After completing the Synthesize and Implementation process, result is compared with actual result of these methods. From the obtained result, calculate the area, power, delay and frequency of these methods and Proved Proposed Delay PWM is provides less area utilization than the other two methods. Then Counter PWM offer high frequency than the any other methods [78].

<table>
<thead>
<tr>
<th>Methods</th>
<th>LUT</th>
<th>Slices</th>
<th>Delay(ns)</th>
<th>Frequency(MHz)</th>
<th>Power(mW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comparator PWM</td>
<td>6</td>
<td>9</td>
<td>3.440</td>
<td>290.723</td>
<td>222</td>
</tr>
<tr>
<td>Counter PWM</td>
<td>3</td>
<td>2</td>
<td>2.733</td>
<td>365.992</td>
<td>275</td>
</tr>
<tr>
<td>Delay PWM</td>
<td>2</td>
<td>2</td>
<td>3.593</td>
<td>278.296</td>
<td>213</td>
</tr>
</tbody>
</table>

Table 3.1 Performance analysis of generation of various PWM Techniques.
Fig. 3.20 Comparison of area and delay between various PWM methods.

Fig. 3.21 Comparison of frequency and power between various PWM methods.
3.6 Conclusion

The design of Comparator PWM, Counter PWM and Delay PWM using Verilog and implemented in a Xilinx Spartan 3 XC3S50 (package: pQ208, speed grade:-5) FPGA using the Xilinx ISE 10.1i design tool. Total equivalent LUT in case of Comparator PWM is 6, Counter PWM is 3 and that is improved to 2 using Delay PWM. The power consumption in the case of Comparator PWM is 222mW, for the Counter PWM it is 365mW and it is improved to 213 using proposed Delay PWM. The number of occupied slices used in Comparator PWM is 9, Counter PWM is 2 and that is improved to 2 using Delay PWM. From the result, concluded that proposed Delay PWM offer low area and less power consumption than the existing Comparator and Counter based PWM. For high speed application counter PWM is better than the other two methods.