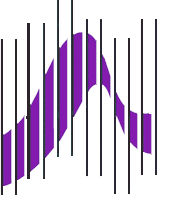
Prithvi Patel – 27890

Mehul Amipara - 25809

|  |
| --- |
| Project Title :- Measurement of the time per Rotation of the Turning Table |



Hochschule Ravensburg – Weingarten

## Guided by :- Dr.-Ing., Professor Walter Ludescher

## Ing. Professor Christoph Weber

**ACKNOWLEDGEMENT**

I am glad to express our indebtedness to our guide Prof. Dr. Walter Ludescher, Department of Electrical Engineering and Information Techology, Hochschule Ravensburg – Weingarten, for your valuable time and guidance. I want to thank you for encouraging me for the execution of the Project task .

I am also thankful to Ing. Prof. Weber, Department of Electrical Engineering and Information Technology, for providing valuable assistance and insight during the experimental process.

Also, Many students, especially my classmates and team member himself, have made valuable suggestions on this proposal which gave us an inspiration to improve our assignment. I am thankful to all the Students who help me in order to improve and execute our Assignment.

**ABSTRACT**

In this Project, We have designed a Circuitry by using VHDL(Very High Speed Integrated Circuit Hardware Description Language) Programming Language. This Project leads to the result of the Measurement of the time per rotation taken by the disk on the Turning Table. So, How it works! Here it goes : First of all, the Sensor that has been used for the Hardware Setup is an Infrared Sensor. The Circuitry gets a pulse, a Raw Sensor Signal, from the Sensor when a piece of block placed on the disk cuts the infrared line of the sensor. The neat or clear Sensor signal is required in order to use it for the counter. The cleaning part has been carried out by one of the blocks in between. The counter calculate the time taken by the disk for a single rotation and has been sent to the host computer by using UART and RS-232 protocol. The Project involves the Circuit designing using VHDL, a Hardware Description Language. Testing of the VHDL code has been done on the Spartan 3E Board. We have designed our Circuitry to make the calculations as accurate as possible. We had put our hard efforts to make it accurate so that for the large scale production of the chips executing this circuitry can be an optimal design.

**Target Specification Status**

This document has been submitted to HS-Weingarten and the features of the ASIC documentation protocols has been taken in measure. I , hereby, assure that the design guidelines provided by Prof. Ludescher, has been followed.

All Project related topics has been described in this document.

**TABLE OF CONTENTS**

List of figures

Few important Key Words

Naming Conventions

Requirement and Objectives of the Project

Rules and Protocols to keep in mind

* Rules and Regulations
* Naming Conventions

**List of Figures**

**Few important keywords**

**Requirement and Objectives of the project**

The objective of this project is to learn how an engineer should design the Circuitry using VHDL programming, how things actually works in the real world. The ultimate target is to find the time of a single rotation taken by the disk on the Turning Table. The FPGA development board that has been used for this project is Spartan 3E FPGA development Board. The VHDL code that has been developed during this project is designed to take the pulse that comes from the sensor and when it detects that pulse, it start counting and increments the counter at every one millisecond. After some milliseconds, around 2000 counts ideal but it depends on the speed of the disk rotating, when it detects the second pulse that’s coming from the sensor, then the counter stops counting and then is been sent to the computer via UART considering RS-232 protocol. Then there exist a C source code that reads the data that’s coming from the port where the output of the FPGA board is connecting, it reads the values and print it on the command line terminal. For testing, GTKterm has been used for the project.

There were certain requirement that has also been taken into consideration.

* The requirement is that the time that has been counted by the counter and then after sent to the transmission block, it should send the data at the Baud rate 9K6 Hz or 9600 Hz.
* Finite State Machine algorithm has been used to prevent latches.
* The whole block should have at least 4 signals : reset, clock, sensor signal, and txd signal.
* Clock speed is required to be 50MHz, but in this design, one more specialty is added, that is, this system can be completely synchronised for 50MHz as well as 10 MHz. There is a switch in between when the design gives results in 50Mhz as well as 10Mhz with using just a switch.
* The transmission should be using UART considering RS-232 protocol.

In the top level, it seems like that there is just a single block that has been doing all the work. But actually, it works quite opposite. It has been divided into multiple small blocks that carries out small tasks and at the end, when all the blocks are connected, it seems like something big is happening.

**Naming Conventions**

**CHAPTER 1 : INTRODUCTION**

There are certain arrangements needed in order to do the design of this Circuitry, such as the Hardware Setup, Software Setup for Stimulation and Software to upload the code to the FPGA board. These things are going to be discussed in this Chapter. In order to upload the code into the FPGA board, a short description is required for the FPGA.

* 1. **Introduction to FPGA**

A Field Programmable Gate Array (FPGA) is a semiconductor device containing programmable logic components and programmable interconnects. The programmable logic components can be programmed to duplicate the functionality of basic logic gates such as AND, OR, NOT, XOR or more complex combinational functions such as decoders or math functions. The FPGA configuration is generally specified using a hardware description language (HDL). The most common FPGA architecture consists of an array of logic blocks called Configurable Logic Block (CLB) or Logic Array Block (LAB), I/O pads- to make off chip connections and programmable routing channels to implement logical functions. FPGAs have analogue features in addition to digital functions. The most common analogue feature is programmable slew rate and drive strength on each output pin. Another relatively common analogue feature is differential comparators on input pins designed to be connected to differential signalling channels.

For more information :- There are many sources available on internet. I have mentioned a few of those!

<https://en.wikipedia.org/wiki/Field-programmable_gate_array>

<https://www.xilinx.com/products/silicon-devices/fpga/what-is-an-fpga.html>

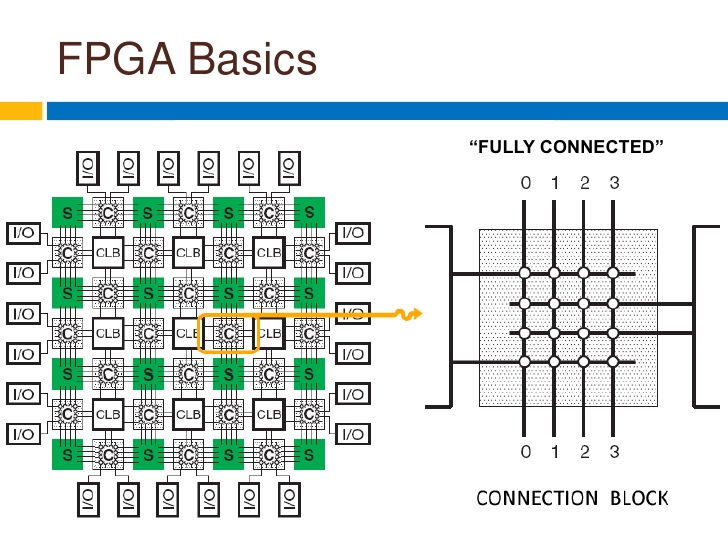


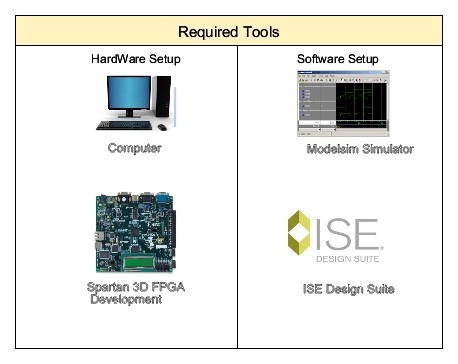
Fig. 1 :- General Architecture of FPGA

* 1. **Introduction to the hardware used for the Project**

The Spartan 3E FPGA development Board provides all the basic features and said to logically optimized.

* For applications where logic densities matte more than I/O count.
* Ideal for logic integration, DSP co-processing and embedded control, requiring significant processing and narrow or few interfaces.

For the project, Spartan 3E : XC3S500E : FG320 has been used.



The Spartan 3E FPGA board comes built in with many peripherals that help in the proper working of the board and also in interfacing the various signals to the board itself. Some of the peripherals are :-

* 2-line, 16 Character LCD screen
* PS/2mouse/keyboard port
* VGA display port
* Two 9-pin RS-232 ports
* 50 MHz clock oscillator
* On-board USB-based FPGA download and debug interface
* Four slide switches and four push-button switches

Moreover, The Spartan-3 family architecture consists of five fundamental programmable functional elements: Configurable Logic Blocks (CLBs) contain RAM-based Look Up Tables (LUTs) to implement logic and storage .Elements that can be used as flip-flops or latches. CLBs can be programmed to perform a wide variety of logical functions as well as to store data. Input/output Blocks (IOBs) control the flow of data between the I/O pins and the internal logic of the device. Each IOB supports bidirectional data flow plus 3-state operation. Twenty-six different signal standards, including eight high performance differential standards, are available. Block RAM provides data storage in the form of 18-Kbit dual-port blocks. Multiplier blocks accept two 18-bit binary numbers as inputs and calculate the product. Digital Clock Manager (DCM) blocks provide self-calibrating, fully digital solutions for distributing, delaying, multiplying, dividing, and phase shifting clock signals.

* 1. **Information about VHDL**

VHDL(VHSIC-Very High Speed Integrated Circuit Hardware Description Language) is a hardware description language used in electronic design automation to describe digital and mixed-signal systems such as field-programmable gate arrays and integrated circuits. VHDL can also be used as a general purpose parallel programming language.

VHDL is commonly used to write text models that describe a logic circuit. Such a model is processed by a synthesis program, only if it is part of the logic design. A simulation program is used to test the logic design using simulation models to represent the logic circuits that interface to the design. This collection of simulation models is commonly called a *testbench*.

A VHDL simulator is typically an event-driven simulator.This means that each transaction is added to an event queue for a specific scheduled time. E.g. if a signal assignment should occur after 1 nanosecond, the event is added to the queue for time +1ns. Zero delay is also allowed, but still needs to be scheduled: for these cases Delta delay is used, which represent an infinitely small time step. The simulation alters between two modes: statement execution, where triggered statements are evaluated, and event processing, where events in the queue are processed.

VHDL has constructs to handle the parallelism inherent in hardware designs, but these constructs (*processes*) differ in syntax from the parallel constructs in Ada (*tasks*). Like Ada, VHDL is strongly typed and is not case sensitive. In order to directly represent operations which are common in hardware, there are many features of VHDL which are not found in Ada, such as an extended set of Boolean operators including **NAND** and **NOR**.

VHDL has file input and output capabilities, and can be used as a general-purpose language for text processing, but files are more commonly used by a simulation testbench for stimulus or verification data. There are some VHDL compilers which build executable binaries. In this case, it might be possible to use VHDL to write a *testbench* to verify the functionality of the design using files on the host computer to define stimuli, to interact with the user, and to compare results with those expected. However, most designers leave this job to the simulator.

It is relatively easy for an inexperienced developer to produce code that simulates successfully but that cannot be synthesized into a real device, or is too large to be practical. One particular pitfall is the accidental production of transparent latches rather than D-type flip-flops as storage elements.

One can design hardware in a VHDL IDE (for FPGA implementation such as Xilinx ISE, Altera Quartus, Synopsys Synplify or Mentor Graphics HDL Designer) to produce the [RTL](https://en.wikipedia.org/wiki/Register-transfer_level) schematic of the desired circuit. After that, the generated schematic can be verified using simulation software which shows the waveforms of inputs and outputs of the circuit after generating the appropriate testbench. To generate an appropriate testbench for a particular circuit or VHDL code, the inputs have to be defined correctly. For example, for clock input, a loop process or an iterative statement is required.[[12]](https://en.wikipedia.org/wiki/VHDL#cite_note-12)

A final point is that when a VHDL model is translated into the "gates and wires" that are mapped onto a programmable logic device such as a CPLD or FPGA, and then it is the actual hardware being configured, rather than the VHDL code being "executed" as if on some form of a processor chip.

<https://en.wikipedia.org/wiki/VHDL>

**CHAPTER 2 : GUIDELINES AND REQUIREMENT**

Everything related to circuit designing with VHDL will be discussed in this Chapter. This chapter includes the design guidelines that has to be done while programming with VHDL. These guidelines are ideal and are currently used worldwide. This chapter also describes the requirements that need to be fulfilled.

* 1. **Design Guidelines**

There are certain guidelines that need to follow :

1. A filename or dirname MUST NOT start with a special char or a number.
2. A dot MUST BE used as a separator between filename and its extension.

Example : content of the file has to be :

mycirc\_e.vhd a VHDL-entity called mycirc\_e

mycirc\_a.vhd a VHDL-entity called mycirc\_a or mycirc\_a1

README.txt some text

1. VHDL directories should be in the Tree Structure. This tree structures of four more directories inside(Documentation, Presentation, gcc and VHDL). The documentation directory consists of a **report.pdf** file(file name can be anything). The Presentation directory should consist a presentation.ppt or .pptx (filename can be anything). The gcc directory consists of the C-code that reads the data coming from the port ttyS0. The VHDL code should have the three more directories, sim – it consists of the simulation work file used in ModelSim to stimulating the code; src – it consists of the all the files related to VHDL program with the top level file.
2. In source files, Each ENTITY has its own file. Each ARCHITECTURE has its own file. And each Architecture has its own TESTBENCH(es).
3. Use of TABS is forbidden.
4. Using more than 80 characters per line is forbidden.
5. For signal-names in entities, the following rule should be applied:

* rb\_i : input, async. Global reset, active low
* cp\_i : input, async. Global System clock, active at rising edge
* xx\_i : input signal named xx\_i
* xx\_o ; output signal named xx\_o
* xy\_io : in-out signal

1. for all signals in Architecture, rules are : xyz\_s : an internal signal named xyz\_s
2. All architectures containing memory must have an input called rb\_i and cp\_i.
3. All architecture containing no memory must not have an input called rb\_i and cp\_i.
4. Constants and components must be defined in a Package

**2.2 Requirements**

There are certain requirements that need to be fulfilled. This requirement section basically consists of Hardware requirements and software requirements.

The Hardware requirements includes :

* A turning table set up for calculating the speed of a single rotation of the Disc.
* An infrared sensors placed on the turning table
* DB9 RS-232 Cable from FPGA board to PC
* System Clock and Asynchronous Reset button
* A DILIGANT Spartan 3E FPGA board for circuit implementation(for counting and measurement)
* UART for data transmission to PC
* PC with UART interface

The Software requirements includes :

* VHDL for Programming FPGA and UART
* C-code for printing the data coming from the port ttyS0 at the baudrate 9600 Hz
* MODELSIM or System Simulation
* ISE for synthesizing the VHDL program and uploading to the FGPA board

**CHAPTER 3 : CIRCUIT DESIGNING WITH VHDL**

A brief overview of the top level of the Circuit blocks will be described in this chapter. Moreover, the top level consists of some small blocks such as counters, multiplexers,etc is going to described in this chapter.

**3.1 Specification and Block diagram**