

Department of Electronic and Telecommunication Engineering

University of Moratuwa, Sri Lanka

EN4603 - Digital IC Design



# Laboratory Experiment 1

## RTL Synthesis

Laboratory Report

Submitted by

Thalagala B.P. 180631J

Submitted on

January 8, 2023

## Contents

List of Figures	1
List of Tables	1
List of Abbreviations	1
1 Introduction	2
1.1 Practical . . . . .	2
1.2 RTL Synthesis . . . . .	2
2 Exercise	3
2.1 System Clocks & Resets . . . . .	3
2.1.1 System Clocks . . . . .	3
Bibliography	3

## List of Figures

1 Overview of an Register-Transfer Level (RTL) synthesis flow[1]. . . . .	2
---	---

## List of Tables

## List of Abbreviations

**GPDK** Generic Process Design Kit

**HDL** Hardware Description Language

**RTL** Register-Transfer Level

*\*PDF is clickable*

### **Note:**

All the materials related to the report can also be found at <https://github.com/bimalka98/Digital-IC-Design>

# 1 Introduction

## 1.1 Practical

In this practical, we will be using *Cadence Genus* to synthesize an example [RTL](#) design, a transceiver. As inputs to Genus, we will provide

1. Source Verilog files
2. Technology libraries provided by the fabrication plant (here, 45 nm educational Generic Process Design Kit ([GPDK](#)) given by Cadence) : (.lib, .lef, .tch)
3. Timing constraints

and will obtain the synthesized netlist (Verilog files) and further timing constraints (.sdc) as output. We will then analyze the area, timing and power of the synthesized design.

## 1.2 RTL Synthesis

In the context of digital hardware design, [RTL](#) synthesis is the process of converting an [RTL](#) description of a digital circuit into an optimized gate-level design.

The [RTL](#) description of a digital circuit is written in a Hardware Description Language ([HDL](#)) such as Verilog or VHDL. It specifies the digital circuit in terms of the flow of digital signals between registers, and the logical operations that are performed on those signals as they are transferred between the registers.

During [RTL](#) synthesis, an [RTL](#) compiler reads the [RTL](#) description of the digital circuit and generates an optimized gate-level representation of the circuit. This gate-level representation is a description of the digital circuit in terms of gates and interconnections between them. Figure 1 illustrates an overview of an [RTL](#) synthesis flow.

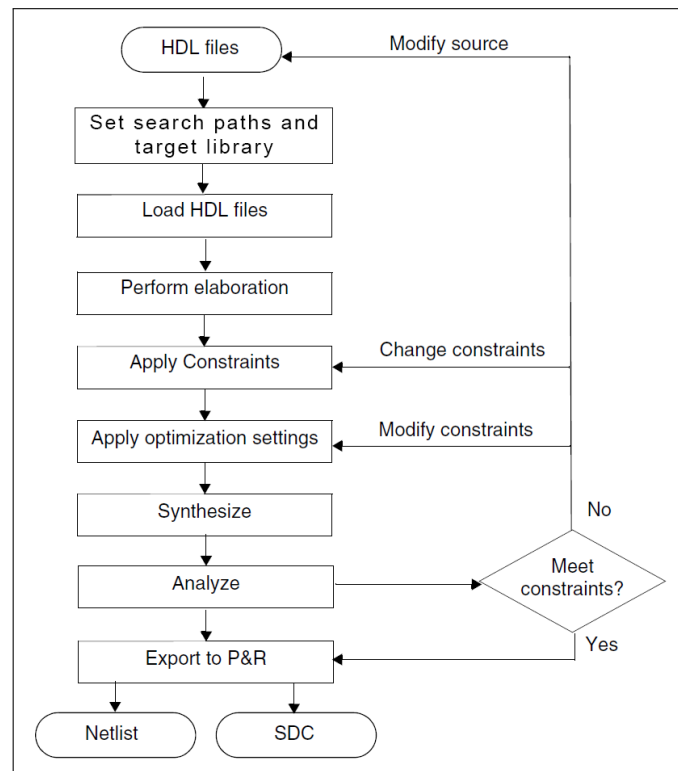


Figure 1: Overview of an [RTL](#) synthesis flow[1].

## 2 Exercise

### 2.1 System Clocks & Resets

#### 2.1.1 System Clocks

---

## Bibliography

- [1] “Genus User Guide for Legacy UI,” version 19.1, November 2019, published by *Cadence Design Systems, Inc.*