University of Moratuwa

System Verilog for ASIC/FPGA Design & Simulation 2023

ASSIGNMENT 4 - FPGA IMPLEMENTATION - REPORT, CODE & VIDEO

MANIMOHAN T.

This report is submitted as Assignment 4 - FPGA Implementation - Report, code & Video

7th May 2023

ABSTRACT

Max pooling is an operation used in neural networks (AI) to reduce the size of an image.

.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 ASSIGNMENT 4 - FPGA IMPLEMENTATION - REPORT, CODE & VIDEO

1. Explain the functionality of the mvm_uart_system rtl and the python script

   In general, RTL (Register Transfer Level) refers to a hardware description language (HDL) representation of a digital circuit at a low-level abstraction. RTL describes the flow of data between registers and operations that can be performed on that data. It is used in digital design and integrated circuit (IC) development.

   The functionality of an RTL module would depend on its specific implementation and purpose. UART (Universal Asynchronous Receiver-Transmitter) is a commonly used communication interface for serial communication between digital devices. If the ”mvm_-uart_system” RTL module is related to UART functionality, it is likely designed to handle UART communication tasks such as sending and receiving data, configuring baud rates, handling start and stop bits, and managing data buffers.

   As for the Python script, it would depend on its specific implementation and purpose as well. Python is a high-level programming language known for its readability and ease of use. It is often used for various tasks, including system programming, automation, and data processing. The script might interact with the ”mvm_uart_system” RTL module, possibly by sending or receiving data through the UART interface, configuring settings, or performing other operations necessary for the specific application.

2. Explain the process(with screen shots) of programming the fpga
   Programming an FPGA (Field-Programmable Gate Array) involves several steps. Here is a high-level overview of the process:

* Design: The first step is to design the desired functionality of the FPGA. This involves creating a hardware description using a hardware description language (HDL) such as VHDL or Verilog. The design includes defining the logic circuits, interconnections, and desired behavior of the FPGA.
* Synthesis: Once the design is complete, it needs to be synthesized into a netlist. Synthesis is the process of converting the high-level design description into a lower-level representation of gates, flip-flops, and other basic elements of the FPGA.
* Optimization: After synthesis, the netlist may undergo optimization to improve performance, reduce power consumption, or meet specific requirements. Various optimizations can be applied, such as technology mapping, resource sharing, and logic minimization techniques.
* Place and Route: The next step is to place and route the synthesized design onto the physical resources of the FPGA. This process determines the physical location of each

logic element and establishes the routing connections between them. It ensures that timing and electrical constraints are met while utilizing the available resources efficiently.

- Bitstream Generation: Once the placement and routing are completed, a bitstream file needs to be generated. The bitstream is a binary file that contains configuration data for the FPGA. It defines the states of each configurable element, such as lookup tables (LUTs), flip-flops, and interconnections.
- Programming the FPGA: The generated bitstream file is then loaded onto the FPGA device. The specific method of programming varies depending on the FPGA and the development environment. It can be done through a JTAG interface, using a programming cable connected to the development board, or through other communication protocols.
- Configuration and Initialization: After the bitstream is loaded onto the FPGA, the configuration process takes place. The FPGA reads the bitstream and configures its internal resources based on the programmed values. Initialization procedures may be necessary to set the initial states of registers and other components before the FPGA starts operating.
- Testing and Debugging: Once the FPGA is programmed and configured, it can be tested and debugged. This involves verifying the functionality and performance of the implemented design by providing appropriate inputs and observing the outputs. Debugging tools and techniques can be utilized to identify and resolve any issues or errors in the design or implementation.

It's important to note that the specific details of each step may vary depending on the FPGA vendor, development tools, and project requirements. FPGA programming requires knowledge of hardware description languages, FPGA architectures, design tools, and debugging techniques.