Homework 6: Report
December 3, 2017
Sabbir Ahmed

1 Background

In this project students will learn to use generated IP Cores: a block RAM and a CORDIC processor for calculating square root. An essential skill to develop is the coding of a control finite state machine (FSM) to interface modules.

1.1 Requirements

- The design should access 4 bytes from the serial port to initiate action
- The implementation should assume 4 bytes will be sent: ADDRESS- HIGH, ADDRESS-LOW, DATA-HIGH, DATA-LOW
- After 4 bytes are received the design should
  - Read the contents of the RAM and send the HIGH-BYTE, then LOW-BYTE back through the UART
  - Send the data bytes through the CORDIC Sqrt processor to compute the result
  - The `sqrt` result just computed should be stored in the RAM at the same address that was just read
- The hardware design should use 115200 baud rate and require NO modification to compile
- The design must include a simulation of the top-level design and discuss it in the report with a parameter to control baud rate. It is strongly recommended to consider modify the baud rate for the purpose of top-level simulation – this should be controlled by a SINGLE parameter set in the test- bench. The design should not modify the UART implementation files.
- The FSM controller must as much as possible rely on use of status and control signals to manage the datapath timing, the design should not rely on "fixed" waits. This represents a better abstraction.

2 Design Approach

The source code for interfacing with UART was provided. The controller FSM was implemented in 7 states visualized in Figure **??**. The module provides all the glue logic required to interface the UART with the `sqrt` module and the block RAM.

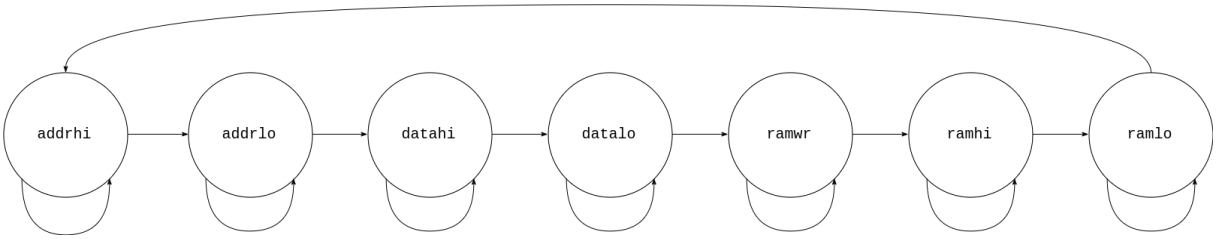

Figure 1: State Diagram of the Implementation of the Controller