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.

Figure 1 provides the datapath of the interface. Students must identify the status and control signals and create a state machine to control them.

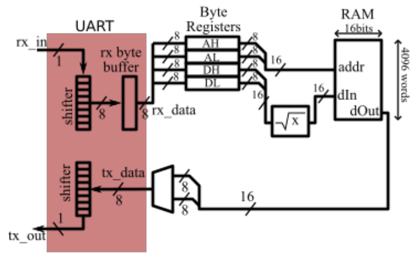

Figure 1: Datapath of the Project

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 to provide all the glue logic required to interface the UART with the `sqrt` module and the block RAM. The multiplexer controlling the `tx_data` in the datapath is integrated into the controller module as the states `ramhi` and `ramlo`. Figure 2 provides the top-level block diagram of the project implementation.

2.1 Controller FSM

The controller FSM consists of 7 states visualized in Figure 3.

2.1.1 addrhi

This is the initial state of the controller to detect when the high bytes of the RAM block address from `rx_data` have been transferred from the UART. The state proceeds to `addrlo` when the shifter at `rx` is empty and `AH` is ready.

2.1.2 addrlo

Activated when the shifter at `rx` is empty and `AH` is ready. This is the second state of the controller to detect when the low bytes of the RAM block address from `rx_data` have been transferred from the UART. The state saves the entire 12-bit address of the RAM block and then proceeds to `datahi` when the shifter at `rx` is empty and `AL` is ready.

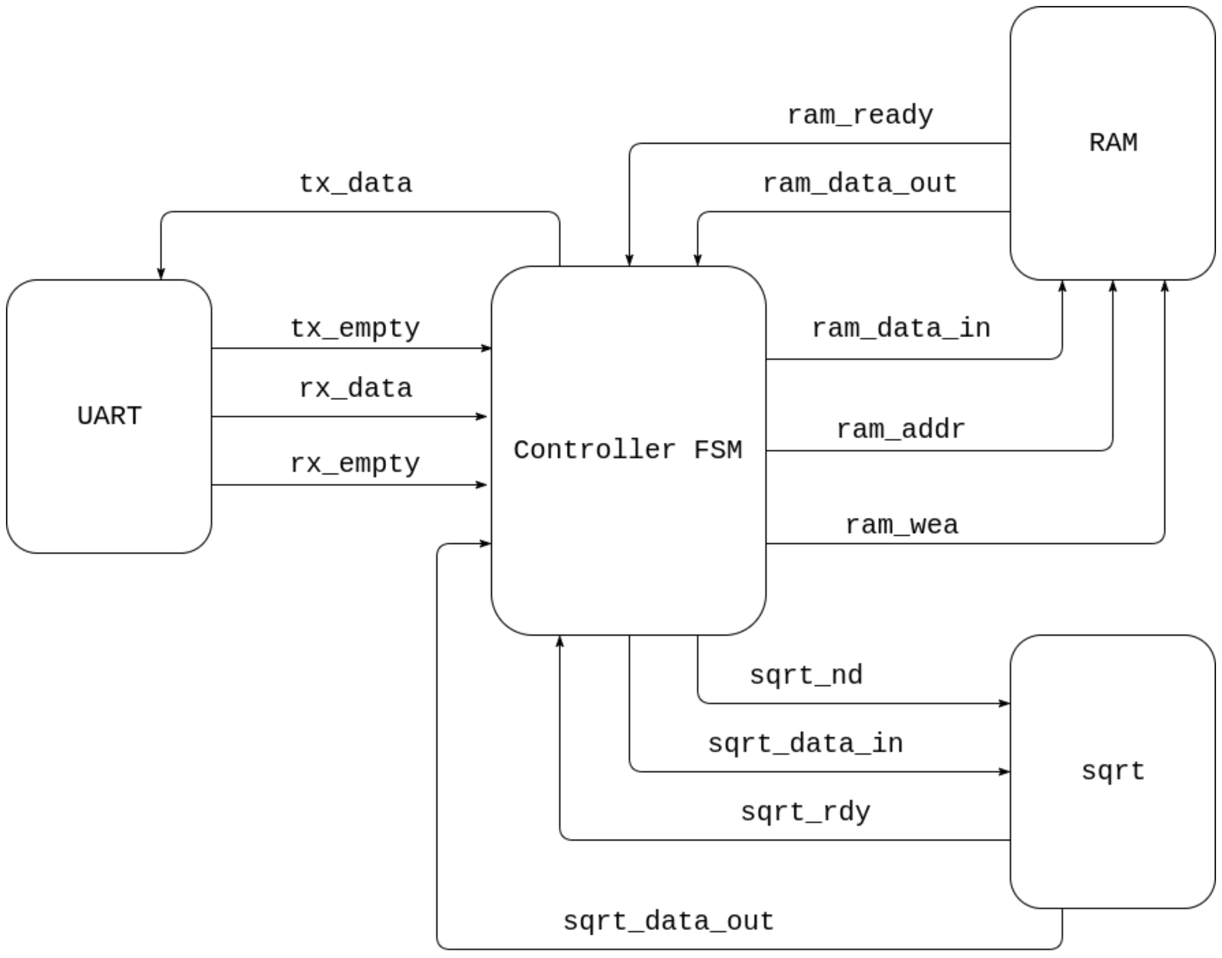

Figure 2: Block Diagram of the Implementation

2.1.3 datahi

Activated when the shifter at `rx` is empty and `AL` is ready. This is the third state of the controller to detect when the high bytes of the RAM block input data from `rx_data` have been transferred from the UART. The state proceeds to `datahi` when the shifter at `rx` is empty and `DH` is ready.

2.1.4 datalo

Activated when the shifter at `rx` is empty and `DH` is ready. This is the fourth state of the controller to detect when the low bytes of the RAM block input data from `rx_data` have been transferred from the UART. The state sends the entire 16-bit data to the `sqrt` module and then proceeds to `ramwr` when the shifter at `rx` is empty and `DL` is ready.

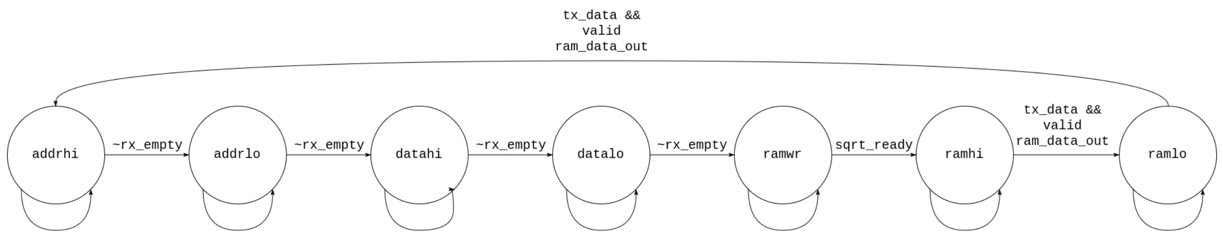

Figure 3: State Diagram of the Implementation of the Controller

2.1.5 ramwr

Activated when the `sqrt` module is and `DL` is ready. This is the fifth state of the controller to write to the RAM block. The state sends the entire 12-bit address and the output of the `sqrt` module to the RAM block and then proceeds to `ramhi`.

2.1.6 ramhi

Activated when the shifter at `tx` is empty and data read from the RAM block is valid and ready. This is the sixth state of the controller to read from the RAM block with the same address and passed back to the UART. The state sends the high 8-bit data read from the `sqrt` module to the `tx_data` in the UART.

2.1.7 ramlo

Activated when the shifter at `tx` is empty and data read from the RAM block is valid and ready. This is the final state of the controller to read from the RAM block with the same address and passed back to the UART. The state sends the low 8-bit data read from the `sqrt` module to the `tx_data` in the UART.

3 Testing

The project submission includes test benches for the controller as well as the top-level module integrating all the modules. Figure 4 shows an example use of the implementation with the memory file contents: 01, 01, 00, 04.

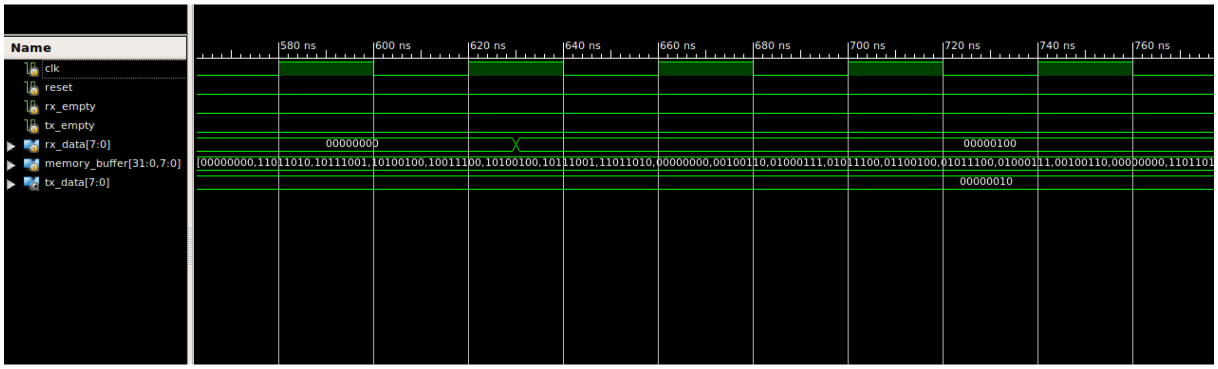

Figure 4: Sample Testbench Waveform of the Controller Module

Figure 4 demonstrates the unit after its final state. The address to the RAM block was passed as 0001 0000 0001 (low 4 bits of the high 01 hex value and the entire low 01 hex value). The data passed to the `sqrt` module was 0000 0000 0000 0100 (00 and 04). The `sqrt` module returns 0000 0010 ($\sqrt{4} = 2$).