8x8 Signed Serial-Parallel Multiplier Project Report

This report outlines the design, implementation, and validation of an 8x8 Signed Serial-Parallel Multiplier using Verilog HDL. The project demonstrates a complete digital system capable of multiplying two 8-bit signed binary numbers, with one input handled serially and the other in parallel.

Project Overview

The 8x8 Signed Serial-Parallel Multiplier (SPM) was implemented on a Basys 3 FPGA board using Xilinx Vivado for synthesis and simulation. The multiplier accepts two 8-bit signed numbers in 2's complement format and produces a 16-bit signed result displayed on the 7-segment display of the FPGA board.
Key Features:

- 8-bit signed multiplicand and multiplier inputs
- Serial-parallel multiplication algorithm
- Complete user interface with switch inputs and 7-segment display outputs
- Scrollable decimal result display
- Operation status indication

Design Architecture

The system architecture follows a modular approach with specialized components handling distinct functions:
Core Modules:

- SPM.v: The main serial-parallel multiplier engine implementing the multiplication algorithm
- Sign_Producer.v: Handles sign calculation for 2's complement multiplication
- Binary_Counter.v and Counter.v: Track the serial input and sequence control1
- BinarytoBCD.v: Converts binary result to decimal for display

User Interface Components:

- Seven_Segment_Display.v: Controls the 7-segment display outputs
- DigitDisplayDriver.v: Manages display data prioritization and timing
- Digit_Selector.v: Handles digit selection for scrollable display
- PushButton.v: Processes button inputs with debouncing

Support Components:

- Clock_Divider.v: Generates appropriate clock frequencies

- Meelay_RisingEdge_Detector.v: Detects button press events
- debouncer.v and synchronizer.v: Eliminate switch bounce and synchronize inputs
- top.v: Top-level module that integrates all components

Implementation Details

Multiplication Algorithm

The serial-parallel multiplication algorithm processes one operand serially while the other is loaded in parallel. The process involves:

1. Loading the parallel input (multiplicand) from switches SW15-SW8
2. Serially processing the multiplier (SW7-SW0) one bit at a time
3. For each multiplier bit, conditionally adding the shifted multiplicand to the partial product
4. Computing the final sign based on the signs of both inputs

The implementation uses a sequential approach that completes the multiplication in multiple clock cycles, similar to the method described in search result.

Hardware Utilization

The design was optimized for the Basys 3 board's Artix-7 FPGA. Key hardware elements included:

- Registers: For storing multiplicand, multiplier, and partial products
- Adders: For partial product accumulation (implemented with CSA.v and HA.v modules)
- Counters: For sequence control and timing
- Display Logic: For driving the 7-segment displays

Timing Considerations

Careful timing analysis was performed to ensure:

- Proper synchronization between serial input processing and result computation
- Accurate debouncing of physical switch inputs
- Appropriate display refresh rates for human readability

Verification and Validation

Simulation Testing

The design was thoroughly tested in simulation using:

1. Waveform Analysis: Verified correct operation of individual modules and the complete system
2. Test Benches: Comprehensive test vectors covering various input combinations
3. Corner Cases: Special attention was given to edge cases such as multiplication by zero or negative numbers

Hardware Validation

Physical testing on the Basys 3 board validated:

1. Functional Verification: Testing with various input combinations
2. User Interface: Validating button responses and display correctness
3. Performance: Measuring maximum operating frequency and multiplication completion time

Contributions:

Ahmed Soliman: Worked on creating serializer and deserializer with the Logisim simulation for the SPM. Designed sub-modules within the SPM module like CSA, TCMP and their submodules. Also, helped in integrating modules responsible for push button detection, clock generation. Worked along with team members in merging the top module and debugging errors.

Hassan Ashraf: I developed the top module, BinarytoBCD, debouncer, synchronizer, 7 segment display, scrolling buttons, and other debugging across the project.

Yousef Elmenshawy: I developed the Logisim simulation for the Signed Sequential Multiplier (SPM) circuit and integrated a 7-segment display to visualize the output. I also designed and implemented the Verilog module for the SPM, ensuring correct handling of signed inputs and sequential operation, and created a comprehensive testbench to verify its functionality. Additionally, I developed the top-level module that connected all components, including push buttons and sign control logic, and thoroughly debugged the design to resolve errors and ensure correct system integration.