The American University in Cairo
Department of Computer Science and Engineering

Final Project Report

8-bit Signed Serial-Parallel Multiplier (SPM)

CSCE 2301 – Digital Design I

Dr . Mohamed Shalan

May 21 , 2025

Team Members:

- Mennatallah Zaid - 90023267
- Maryam Abdelhakem - 900234018

1. Project Overview

This project implements an **8-bit Signed Serial-Parallel Multiplier (SPM)** using **Logisim Evolution** and deploys it on the **Basys 3 FPGA** board. The design combines the hardware efficiency of serial processing with the speed of parallel data input, making it ideal for compact and signed 8-bit binary multiplication. The result is displayed in decimal using a 7-segment scrollable display interface.

---

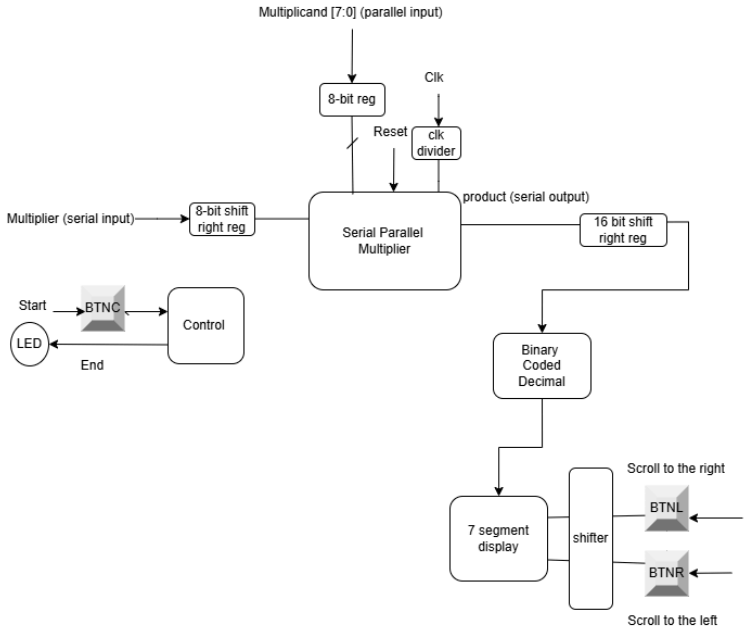
2. System Architecture

2.1 Inputs & Controls

- **Multiplier (8-bit, signed)**: `SW7–SW0`
- **Multiplicand (8-bit, signed)**: `SW15–SW8`
- **BTNC**: Starts the multiplication
- **BTNL/BTNR**: Scroll through result on 7-segment display
- **LD0**: Indicates computation complete

2.2 Block Diagram

The block diagram illustrates the operation of an 8x8 Signed Serial-Parallel Multiplier system designed for FPGA implementation. The user inputs two signed 8-bit binary numbers using toggle switches: the Multiplier (SW7–SW0) is fed serially into an 8-bit shift-right register, while the Multiplicand (SW15–SW8) is loaded in parallel into an 8-bit register. Upon pressing the BTNC (center button), the Control Unit initiates the multiplication process by coordinating data flow and timing through the system, using a Clock Divider to slow down the clock for visible operations. The Serial-Parallel Multiplier (SPM) receives the serial bits from the multiplier register and the parallel multiplicand, producing the 16-bit signed product as a serial output, which is collected in a 16-bit shift-right register. Once complete, the LED (LD0) is turned on to indicate the end of computation. The 16-bit product is then converted from binary to BCD format by a dedicated converter for display purposes. The result is passed to a shifter that controls the scrolling of the 5-digit decimal product through the rightmost three digits of the 7-segment display, while the leftmost digit displays the sign (negative sign indicated by lighting segment G). The BTNL and BTNR buttons allow the user to scroll left and right to view the full result.



Serial Parallel Multiplier:

Figure 4. Bit Serial (Carry Save) Adder

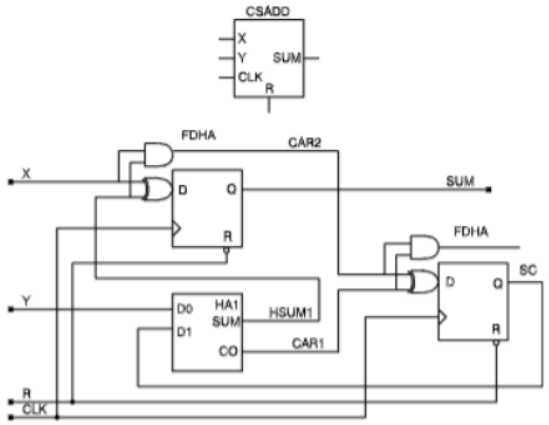


Figure 5. Two's Complement

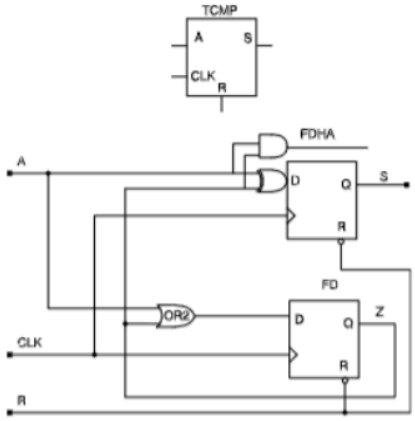
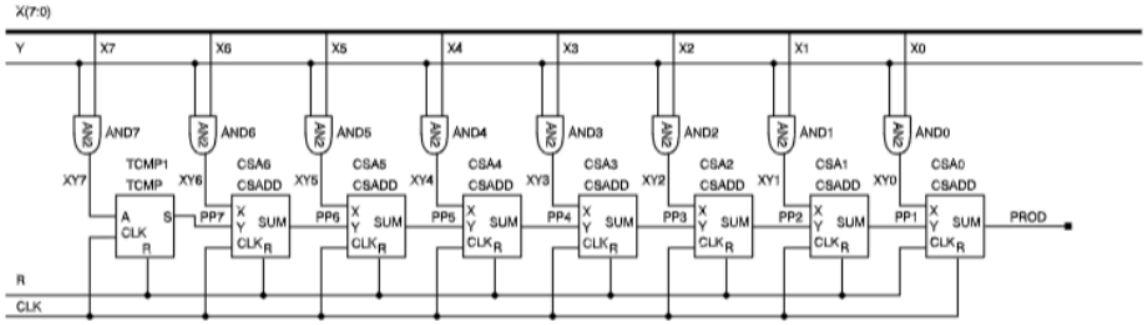


Figure 6. 8-bit Serial-parallel Multiplier



- **Control Unit**:
  Handles data load, clock timing, shifting, and computation status.

- **Serial-Parallel Multiplier (SPM)**:
  Performs signed multiplication (8*8) over multiple cycles (16).

- **Binary to BCD Converter**:
  Uses the **Double Dabble algorithm** for decimal conversion.

- **7-Segment Display Handler**:
  Scrollable output showing:

  - **Sign** (segment G on for negative)
  - **Magnitude** (up to 5 decimal digits)

3. LOGISM

3.1 Logisim (spm.circ)

1. **Reset** the system.
2. Set **Multiplier** (SW7–SW0) and **Multiplicand** (SW15–SW8).
3. Turn **Load = ON** (enables parallel load of multiplier).
4. Pulse **manual clock once**.
5. Set **Load = OFF**.
6. Start **automatic clock** (press `Ctrl+K`).
7. Wait until **LD0 LED turns ON** (multiplication complete).
8. Stop clock (`Ctrl+K`) to prevent result overwrite.

3.2 Display (finalsevensegment.circ)

1. Input the **16-bit signed binary result**.
2. Use **BTNL** and **BTNR** to scroll across the 5-digit output.
3. View sign and magnitude:
    - Leftmost digit: Sign
    - Rightmost three: Scrollable digits

---

4. Implementation Challenges

- **Signed Arithmetic Handling**:
  Proper two's complement logic across registers and BCD conversion.

- **Clock Control & Timing**:
  Managing auto clocks without data corruption.

- **Display Overflow**:
  Scrolling logic built to manage overflow beyond 3 visible digits.

- **FPGA Deployment**:
  Adjusting control signals for compatibility with Basys 3 platform.

---

6. Validation & Testing

6.1 Functional Tests

- Multiplication verified for both positive and negative inputs.

- Manual verification against software-calculated values.

6.2 Edge Case Examples

- `(-128 × -128) = 16384`

- `(127 × -1) = -127`

- `(-1 × -1) = 1`

6.3 Display Validation

- Correct decimal output on 7-segment displays.

- Sign indication with segment G.

- Smooth horizontal scrolling for 5-digit outputs.

---

7. Individual Contributions

Member	Contributions
Maryam Abdelhakim	Built control unit for multiplier, managed clock logic, and led FPGA testing and Debugging
Mennatallah Zaid	Built BCD, Developed 7-segment display circuit using Double Dabble and scrolling buttons
Combined Effort	Built Control Unit for the buttons, Designed Multipler, Connected the modules, Handled edge cases validation and report writing.

---

8. References

- **Electronics StackExchange** – "16-bit to Decimal Conversion": https://electronics.stackexchange.com/questions/713730
- **Double Dabble Algorithm** – Wikipedia: https://en.wikipedia.org/wiki/Double_dabble
- **Microchip App Note** – Binary to BCD: https://www.microchip.com/content/dam/mchp/documents/OTH/ApplicationNotes/ApplicationNotes/DOC0529.PDF
- https://github.com/Manarabdelaty/Fault-SPM

- **Github repo link**: https://github.com/Maryam-Gamal/8-bit-SPM.git

---