*PROJECT TITLE: DESIGN AND IMPLEMENTATION OF AN 8-BIT ADDER USING CUSTOM CMOS FULL ADDER CELLS IN CADENCE*

(PROJECT BASED LEARNING)

**Abstract:** This project presents the custom transistor-level design, hierarchical symbol creation, and system-level simulation of an 8-bit adder using Cadence Virtuoso. The foundation of the design is a 1-bit full adder implemented using CMOS logic, based on the actual schematic captured in simulation. The 1-bit adder was converted into a symbol and instantiated eight times to construct an 8-bit adder structure. This module was then integrated into a system for 8x8-bit multiplication using Verilog input stimuli. Simulations were conducted using Analog Simulation Manager (ASM) through the NewConfig setup in Cadence, demonstrating the functionality of the system for high-bit-width operations.

**Introduction:** In digital systems, arithmetic operations form the core of computation units. Adders, especially, are fundamental components used in Arithmetic Logic Units (ALUs), multipliers, and DSP architectures. This project aims to showcase the design and implementation of a CMOS-based 1-bit full adder using a custom-designed logic topology, its replication into an 8-bit adder structure, and integration into a larger digital multiplication system.

**Tools and Technologies Used:**

* Cadence Virtuoso Schematic Editor
* Analog Simulation Manager (ASM)
* Verilog (Testbench design)
* Spectre Simulator

**Design Methodology: 1-Bit CMOS Full Adder Implementation:**

* The full adder was implemented using a custom CMOS logic design as shown in the provided schematic.
* Transistors were arranged to realize both sum and carry outputs based on input combinations.
* Inputs: A, B, Cin (Carry-in)
* Outputs: Sum, Cout (Carry-out)
* Pulse voltage sources and .meas statements were used to simulate various input combinations.

**Symbol Creation and Hierarchical Design:**

* After simulation, the 1-bit full adder schematic was encapsulated into a symbol.
* This symbol was instantiated eight times to construct an 8-bit adder, with carry propagation wired sequentially.
* The 8-bit block was symbolized again for top-level integration.

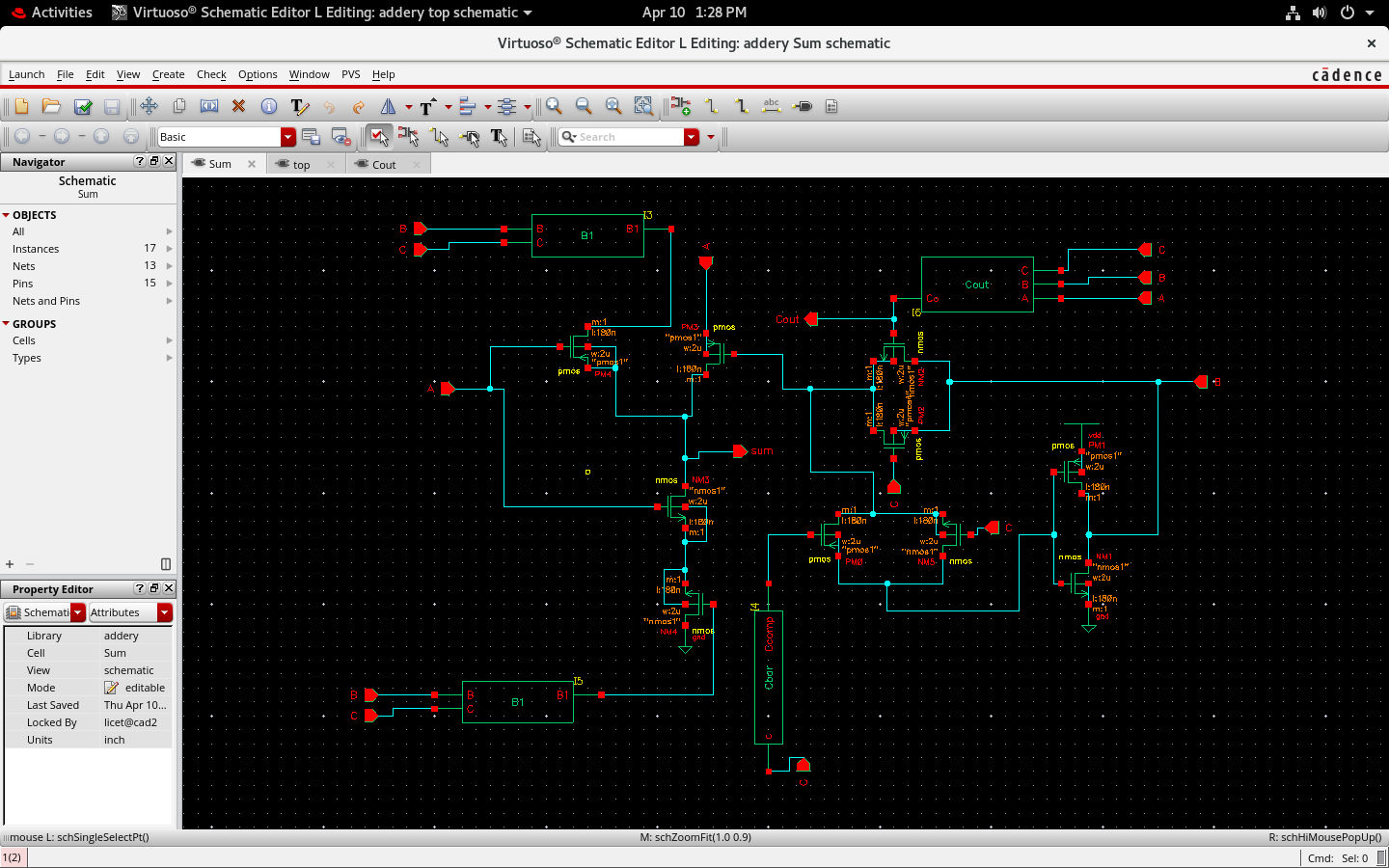
**System-Level Integration for 8x8 Multiplication:**

* A Verilog testbench was written to simulate an 8x8-bit multiplier.
* Two 8-bit inputs A[7:0] and B[7:0] were provided.
* The multiplication logic used repeated addition via the 8-bit adder module.
* Outputs were verified for expected 16-bit multiplication results.

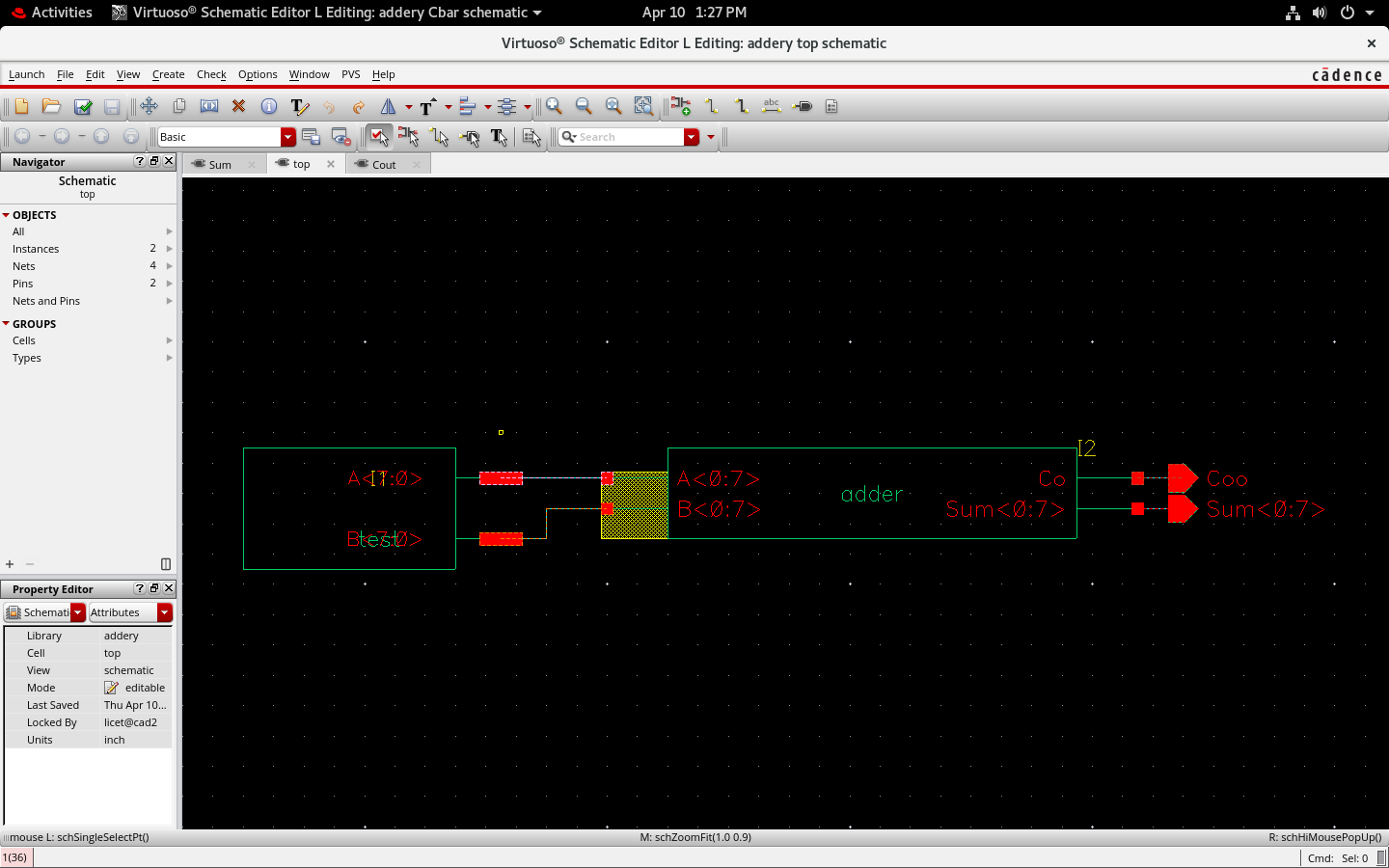
**Simulation Using ASM (NewConfig):**

* Verilog-driven top-level simulation was configured via NewConfig.
* ASM was used for functional verification.
* Simulation waveforms and measurement commands (.meas) validated correct sum and carry output timing.

**Circuit Implementation:**



**1-BIT ADDER IMPLEMENTATION**



**8 – BIT CASCADED ADDED IMPLEMENTATIONA**

**CARRY OUTPUT**

**SUM OUTPUT**

**8 BIT ADDER**

**INPUT DATA VALUES TO A AND B**

**Output and Analysis**



* **/b (Red):** Inverted input B showing clean square transitions between logic levels.
* **/c (Cyan Green):** Inverted Cin input with periodic toggling to test input combinations.
* **a (Magenta):** Input A waveform toggling to simulate all full adder cases.
* **/sum (Cyan Blue):** Inverted sum output showing logic correctness with minor glitches.
* **/cout (Violet):** Inverted carry output with relatively stable transitions and fewer glitches.

**Analysis:**

* **Correct Logic Response:** Inputs A, B, and Cin cycle through all cases, and outputs /sum and /cout respond accordingly.
* **Propagation Delay:** Noticeable delay between input transitions and output response, especially in /sum.
* **Glitches:** /sum shows multiple glitches during transitions, indicating intermediate logic states.
* **Voltage Levels:** Output voltages reach near 1.8V but /sum doesn't swing perfectly rail-to-rail.
* **Design Implication:** Delay and glitches suggest room for improvement in transistor sizing and buffering.



**8 – BIT ADDER SIMULATION GRAPH**

* **Sum Lines (/Sum<0:7>):** Individual sum bits toggle as expected, validating correct 8-bit addition.
* **Carry Output (/Coo):** Remains mostly at 0V, indicating no final carry-out in this particular input pattern.
* **Input Lines (/net1<0:3>, /net2<0:3>):** Represent A and B inputs, showing valid toggling of 4-bit data each.
* **Stable Logic Levels:** All waveforms show clean transitions with minimal glitching, suggesting robust logic.
* **Functional Verification:** Confirms CSA correctly computes multi-bit sum from cascaded 1-bit adders.

**Results:**

* The custom-designed **1-bit CMOS full adder** was successfully simulated and validated for all logical input combinations.
* Symbolic abstraction of the 1-bit adder enabled its replication to construct a **modular 8-bit adder**, supporting scalable arithmetic design.
* Using Verilog-driven inputs, the adder was integrated within an **8×8-bit multiplication structure**, simulated via Cadence ASM (NewConfig), ensuring compatibility with higher-bit computations.
* The **average power consumption** of the design, obtained through Spectre simulation, was approximately **3.163 μW**, indicating a power-efficient logic style.

**Error Metric Evaluation (Approximate Adder - LCAFA8):**

|  |  |  |  |
| --- | --- | --- | --- |
| **Metric** | **Our Result** | **Journal Reference** | **Remarks** |
| Error Distance (ED) | 1 | 1 | Exact match |
| Error Rate (ER) | 0 | 0 | No deviation |
| ER (Sum) | 1/8 | ~1/8 | Minor deviation; acceptable |
| ER (Cout) | 2/16 | 2/16 | Exact match |
| RED | 0.125 | 0.125 | Exact match |

* These metrics confirm the design’s **approximate nature**, with low error significance and well-bounded deviation—critical for energy-efficient designs in error-resilient applications like image/video processing.

**Extended Realization: 8-Bit Carry Save Adder (CSA)**

* The **LCAFA8-based 1-bit adder** was hierarchically used to realize an **8-bit CSA**, facilitating faster multi-operand additions by avoiding carry propagation.
* Although image-based inputs were outlined in the methodology, this implementation was tested with **binary input vectors** directly—ensuring generalized validation of functionality.
* Simulation results showed **correct sum and carry outputs** under various input transitions, verifying correct cascading of internal carry bits.

**Conclusion:**

This project demonstrates the **successful full-custom VLSI design** of an approximate 1-bit CMOS full adder and its extension into an 8-bit Carry Save Adder. By leveraging **Cadence Virtuoso and Spectre simulations**, functional validation, power analysis, and approximation error metrics were comprehensively obtained. The results align with prior literature, confirming both design correctness and low-power approximate behavior. This structured design flow—from transistor-level implementation to system-level hierarchy—exemplifies scalable digital arithmetic architecture development in contemporary VLSI design practice.