MINOR PROJECT REPORT

Design And Simulation of SAP-1 Architecture Using Logisim Evolution.

A Minor Project report Submitted for the Evaluation and Partial Fulfillment of the Requirement for the degree of

Integrated BTech. MTech in ELECTRONICS AND COMMUNICATION ENGINEERING

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Submitted by :

Arthansh Maurya 225/IEC/001

Deepti Kaushik 225/IEC/002

Harshit Arya 215/UEC/049

Supervisor: Dr. Annu Singh

OVERVIEW

- Vision & Objective
- SAP 1 Architecture
- Methodology
- Implementation
- Simulation Result
- Conclusion

CONCEPT: BRIDGING THEORY AND REALITY

The Black Box of Modern Era

- Billions of transistors.
- Complex Pipelining & Cache prediction.
- Challenge: Impossible to visualize bit-level data flow for a beginner

The Academic Standard (Intel 8085)

- Teaches Assembly Language & Pin Diagrams.
- Limitation: We study the pins, but the internal bus movement is still hidden inside the chip.

Our Solution (SAP-1 Implementation)

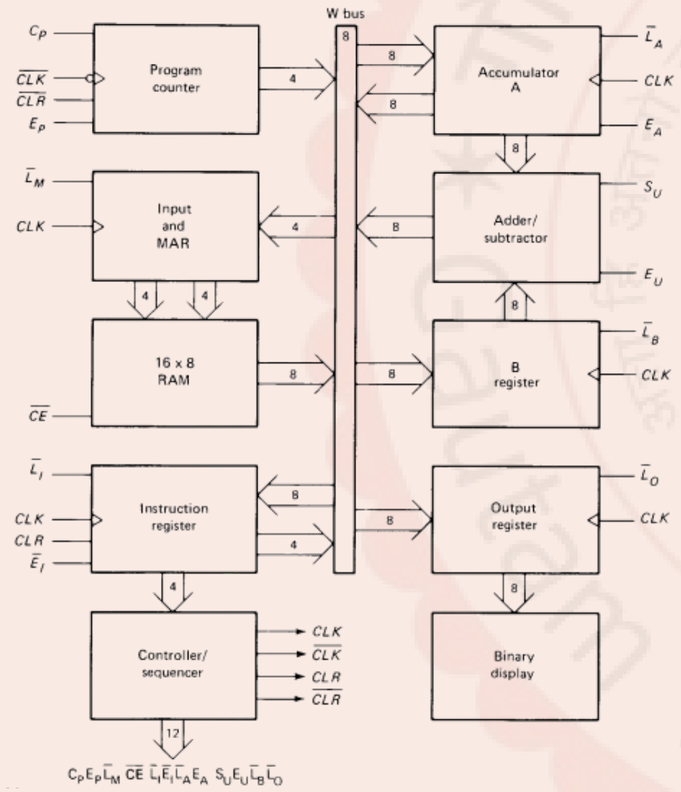
- The "X-Ray" View: We literally "open up" the chip.
- It demonstrates the exact Micro-operations that happen inside an 8085 during a Fetch Cycle.

Conclusion: SAP-1 provides the foundational logic required to understand 8085 and modern architectures.

ARCHITECTURAL EVOLUTION

Adapting SAP-1 for Simulation

Theoretical Model (Malvino)



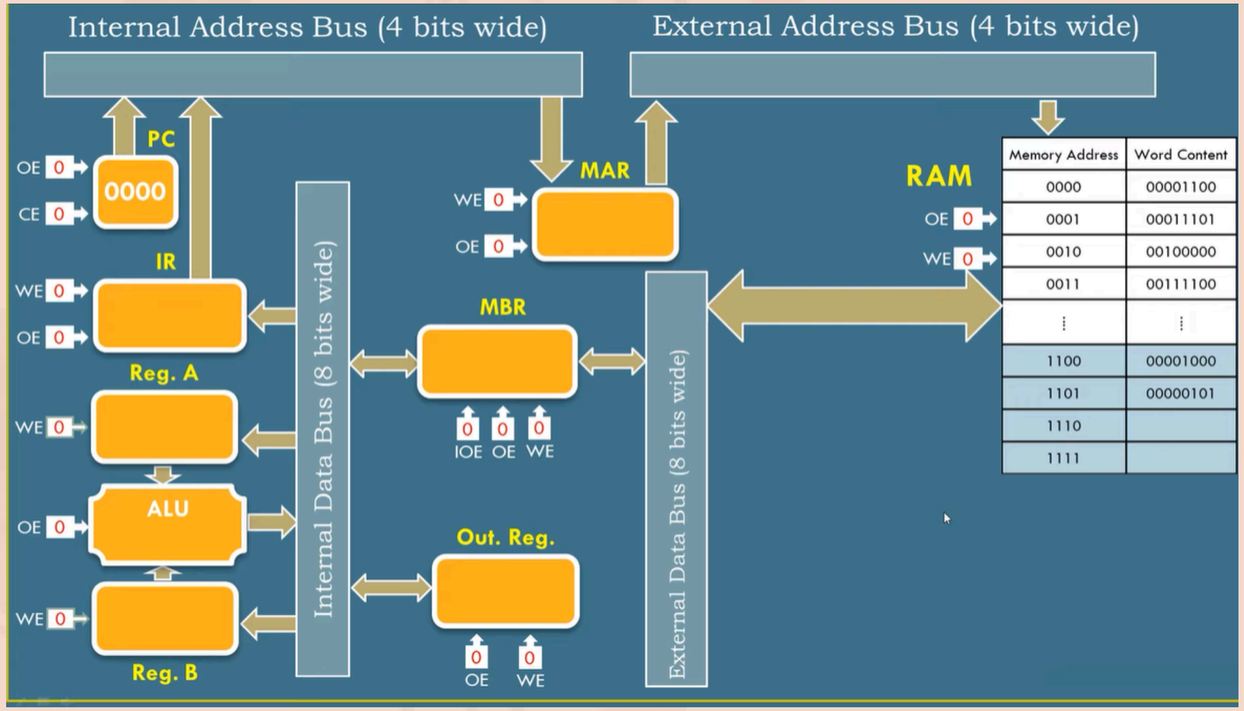
- Single shared W-Bus for all data transfer.
- RAM connects directly to the bus.
- Relies on perfect timing of Output Enable (OE) signals.
- Limitation: In Logisim, direct RAM connection often causes "Bus Contention" (Short Circuits) during switching.

ARCHITECTURAL EVOLUTION

Adapting SAP-1 for Simulation

- Key Innovation: Memory Buffer Register (MBR).
- Split Architecture:
  1.)External Bus: Handles raw RAM data.
  2.)Internal Bus: Handles CPU Register transfers
- Why? MBR act as a “Gateway” fetching data from RAM and safely places it on the CPU bus only when needed.
- Result: 1) Zero Bus Contention 2) Stable Simulation.

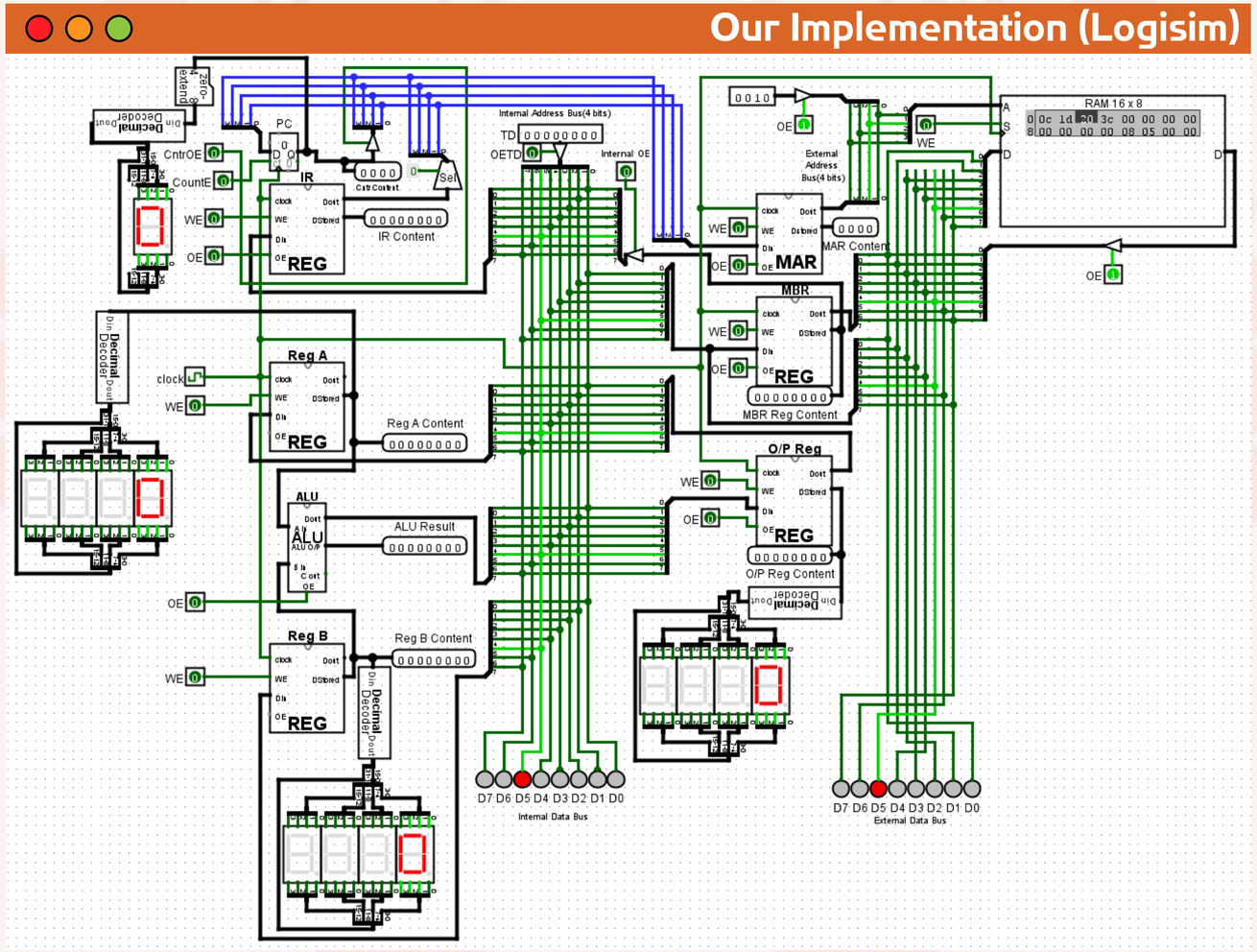
Our Implementation (Logisim)



ARCHITECTURAL EVOLUTION

Adapting SAP-1 for Simulation

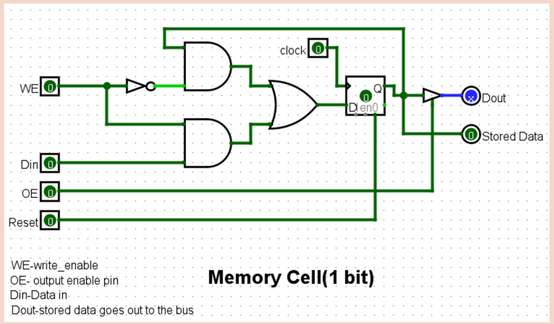
Our Implementation (Logisim)



METHODOLOGY

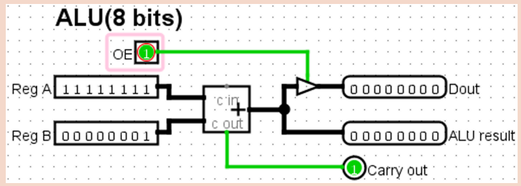
Bottom-Up Design Approach

The Foundation: 1-Bit Memory Cell



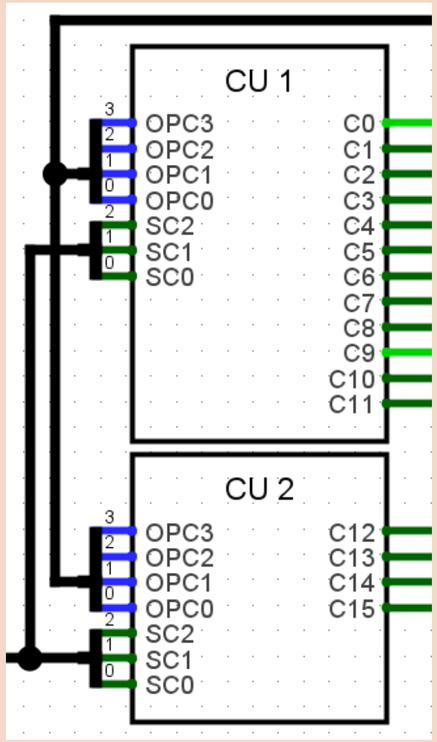
- **Design: Built using D-Flip Flops and Tri-state buffers.**
- **Function: Stores a single bit; forms the building block for all Registers (A, B, IR).**

The Computing Engine: ALU

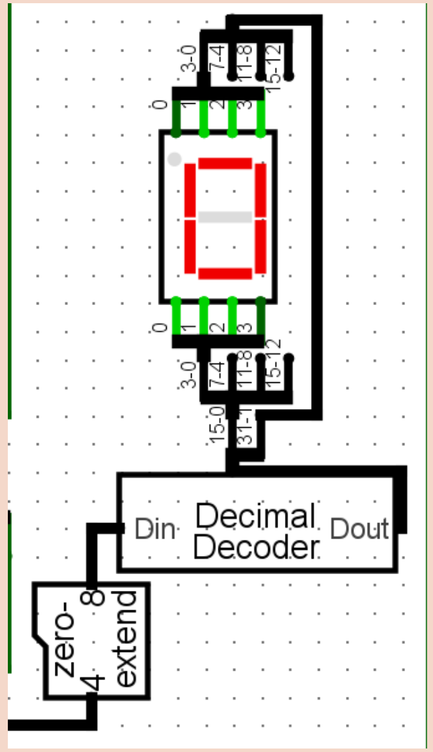


- **Design: 8-Bit Ripple Carry Adder.**
- **Function: Performs binary addition (A + B) and handles Carry flags**

The Control Unit: Hardwired Logic



Visualization: Decimal Decoders



SYSTEM VERIFICATION

From Manual to Automated

Manual T-State Verification

- The “Slow” Test
- Objective: Validate micro-operations clock-by-clock.
- Result: Confirmed correct routing through the MBR buffer.

- Fetch Cycle: Confirmed Program Counter increment and Instruction Latch.
- Execute Cycle: Verified control signals for LDA, ADD, and STORE

Full Program Execution

- The “Fast” Test
- Objective: To self test the architecture using CU.
- Outcome: The Output Register successfully displayed 15 on the 7-segment display.

- LDA 0cH => Reg A loads 05
- LDB 1dH => Reg A loads 08.
- ADD 20H => ALU computes 05 + 03 = 08.
- OUT => Result displayed.

CONCLUSION & FUTURE HORIZONS

Project Conclusion:

- Successful Implementation.
- Operational Verification validating our custom MBR architecture.
- Educational Value: This project provides a transparent, "X-ray" view of CPU internals.

Future Scope:

- Implementing the SAP-2 Architecture.
- CU Upgrade: Replacing the hardwired logic with a Micro-programmed Control Unit (ROM).
- Hardware Realization: Implementing on FPGA for educational puposes.

THANK YOU

For your attention