Design Specifications

System Architecture

- **2-to-4 Row Decoder:** Converts 2-bit address to 4 word lines
- **4x4 SRAM Array:** 16-bit storage organized as 4 rows of 4 bits each
- **Control Signals:** Enable, write enable, and address selection
- **Data Path:** 4-bit input/output interface

Interface Specifications

Inputs:

- `row_address[1:0]` - 2-bit address for row selection
- `enable` - Active-high enable signal
- `write_enable` - Active-high write control
- `data_in[3:0]` - 4-bit data input

Outputs:

- `data_out[3:0]` - 4-bit data output
- `word_lines_debug[3:0]` - Debug visibility of active word lines

Memory Initialization

- Row 0: `4'b0001`
- Row 1: `4'b0010`
- Row 2: `4'b0100`
- Row 3: `4'b1000`

Design Iterations

Iteration 1

Prompt:

```
Create a Verilog design for a 2-to-4 decoder connected to a 4x4 memory
array.
Specifications:
- one module 2-to-4 decoder takes 2-bit address and produces 4 word lines
- one module 4x4 memory array stores 4 bits per row (total 16 bits)
- Top-level module ports:
  - Input: address (2-bit), enable, write_enable, data_in (4-bit)
  - Output: data_out (4-bit)
Initialize the memory with test values: Row 0=0001, Row 1=0010, Row 2=0100,
Row 3=1000.
```

Response: Generated a well-structured design with three modules and generally correct logic implementation.

Simulation Results:

- **Tests Passed:** 8/14
- **Tests Failed:** 6/14
- **Status:** FAILED

Critical Issues Identified:

1. **Enable Control Malfunction:** Word lines remained active when `enable=0`
   - Expected: `word_lines = 4'b0000`
   - Actual: `word_lines = 4'b0001`
2. **Write Operation Display Error:** During writes, `data_out` showed incorrect values
   - Expected: Display new data being written
   - Actual: Displayed old or random data (value `8`)
3. **System Disable Failure:** Final disable test failed due to persistent word line activation

Iteration 2

Refined Prompt:

```
Create a Verilog design for a 2-to-4 decoder connected to a 4x4 memory
array.
Specifications:
- one module 2-to-4 decoder takes 2-bit address and produces 4 word lines
- one module 4x4 memory array stores 4 bits per row (total 16 bits)
- Top-level module ports:
  - Input: row_address (2-bit), enable, write_enable, data_in (4-bit)
  - Output: data_out (4-bit) word_lines_debug (4 bit)

When enable=0, word_lines should be 4'b0000, not active.

Initialize the memory with test values: Row 0=0001, Row 1=0010, Row 2=0100,
Row 3=1000.
```

Response: Implemented proper enable control logic, fixing the primary control signal issue.

Simulation Results:

- **Tests Passed:** 10/14
- **Tests Failed:** 4/14
- **Status:** IMPROVED

Issues Resolved: Enable Control Fixed: System properly disables when `enable=0`

Remaining Issues:

1. **Write Operation Display:** During write operations, `data_out` still showed old stored data instead of new input data
   - Test 6: Expected to show `A`, got `1` (old data)
   - Test 7: Expected to show `B`, got `2` (old data)
   - Test 8: Expected to show `C`, got `4` (old data)

- Test 9: Expected to show `D`, got `A` (inconsistent)

Iteration 3 (Final)

Optimized Prompt:

```
Create a Verilog design for a 2-to-4 decoder connected to a 4x4 memory
array.
Specifications:
- one module 2-to-4 decoder takes 2-bit address and produces 4 word lines
- one module 4x4 memory array stores 4 bits per row (total 16 bits)
- Top-level module ports:
  - Input: row_address (2-bit), enable, write_enable, data_in (4-bit)
  - Output: data_out (4-bit) word_lines_debug (4 bit)

When enable=0, word_lines should be 4'b0000.

Fix: During write operations (write_enable=1), data_out should show the
data being written
(data_in), not the old stored data.

Initialize the memory with test values: Row 0=0001, Row 1=0010, Row 2=0100,
Row 3=1000.
```

Response: Generated perfect implementation with correct write operation behavior.
Simulation Results:

- **Tests Passed:** 14/14
- **Tests Failed:** 0/14
- **Status:** SUCCESS

All Issues Resolved

Final Test Results:

```
Test 1: System disabled - PASS
Test 2-5: Read initial data from each address - ALL PASS
Test 6-9: Write new data to each address - ALL PASS
Test 10-13: Read back written data - ALL PASS
Test 14: Final disable test - PASS

*** ALL TESTS PASSED! DESIGN IS CORRECT ***
```

Results Summary

Iteration	Primary Focus	Tests Passed	Tests Failed	Key Issues Addressed
1	Initial Implementation	8/14 (57%)	6/14 (43%)	Basic structure, multiple control issues
2	Enable Control Fix	10/14 (71%)	4/14 (29%)	Enable signal handling
3	Write Behaviour Fix	14/14 (100%)	0/14 (0%)	Write operation display logic

Performance Metrics

- **Total Iterations Required:** 3
- **Final Success Rate:** 100%
- **Primary Challenge:** Control signal logic (enable and write behaviour)
- **Design Complexity:** Moderate (3 interconnected modules)

Verilog Command used:

iverilog -o design.vvp design.v design_tb.v && vvp design.vvp

Conclusions

1. Systematic Testing Approach

- Comprehensive test suite (14 test cases) enabled precise issue identification
- Clear pass/fail criteria provided unambiguous feedback
- Debug outputs (`word_lines_debug`) enhanced visibility into system behaviour

2. Iterative Prompt Refinement

- **Iteration 1:** Broad functional specification
- **Iteration 2:** Targeted enable control fix
- **Iteration 3:** Specific write behaviour correction

3. Issue Prioritization

- Control logic problems addressed before operational refinements
- Critical functionality (enable/disable) fixed before behavioral improvements