Qwen3-4B-Thinking MLX Benchmark Report
Baseline vs Fine-tuned (LoRA Adapter)

Hardware: MacBook Air M1 (8GB RAM)
Model: Qwen3-4B-Thinking-2507-MLX-4bit
Adapter: LoRA fine-tuned for EDA/Verilog

Date: 2025-12-09 00:34

Generated by Pablo's Benchmark Script

EXECUTIVE SUMMARY
==========

Test Environment:
- Hardware: MacBook Air M1, 8GB RAM
- Model: Qwen3-4B-Thinking-2507-MLX-4bit (4-bit quantized)
- Adapter: LoRA adapter trained on EDA/Verilog dataset
- Framework: MLX (Apple Silicon optimized)

Key Findings:

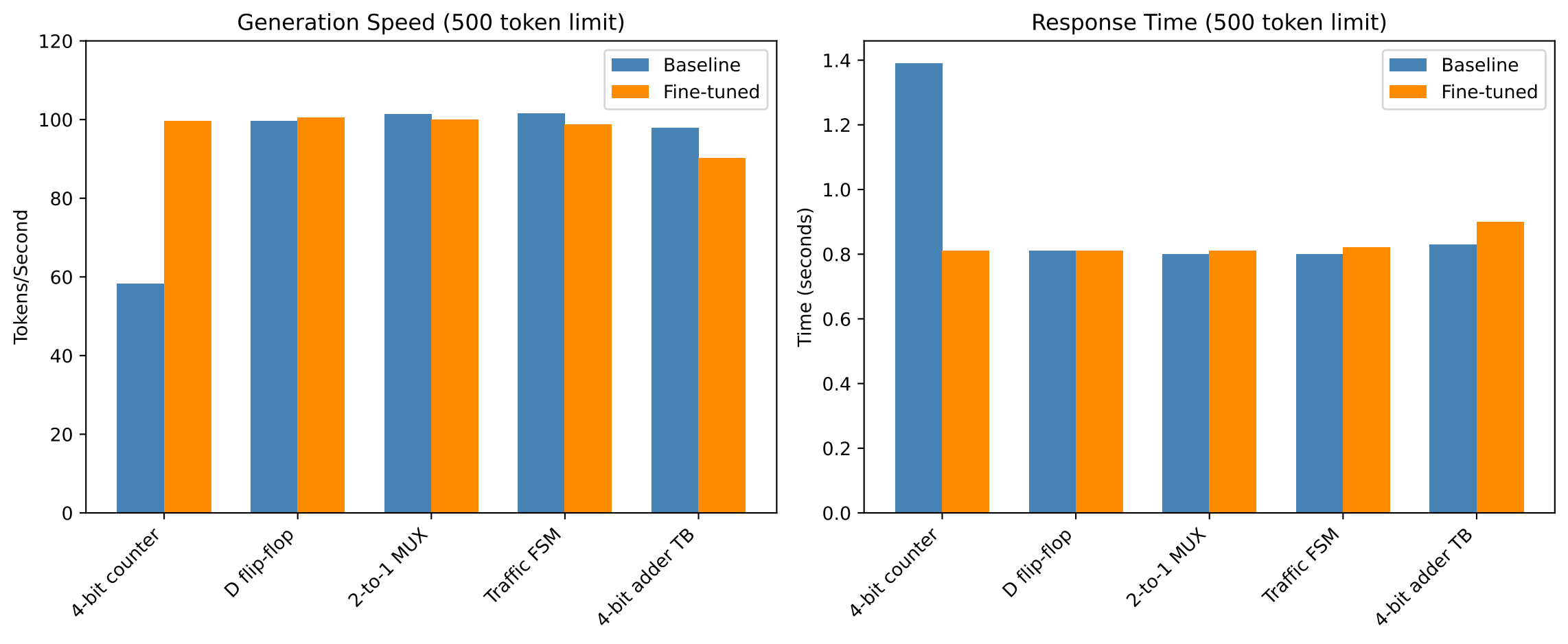
1. PERFORMANCE (Tokens/Second):
   - Short generations (500 tokens): Similar performance
     * Baseline avg: 91.70 tok/s
     * Fine-tuned avg: 97.79 tok/s (+6.6%)

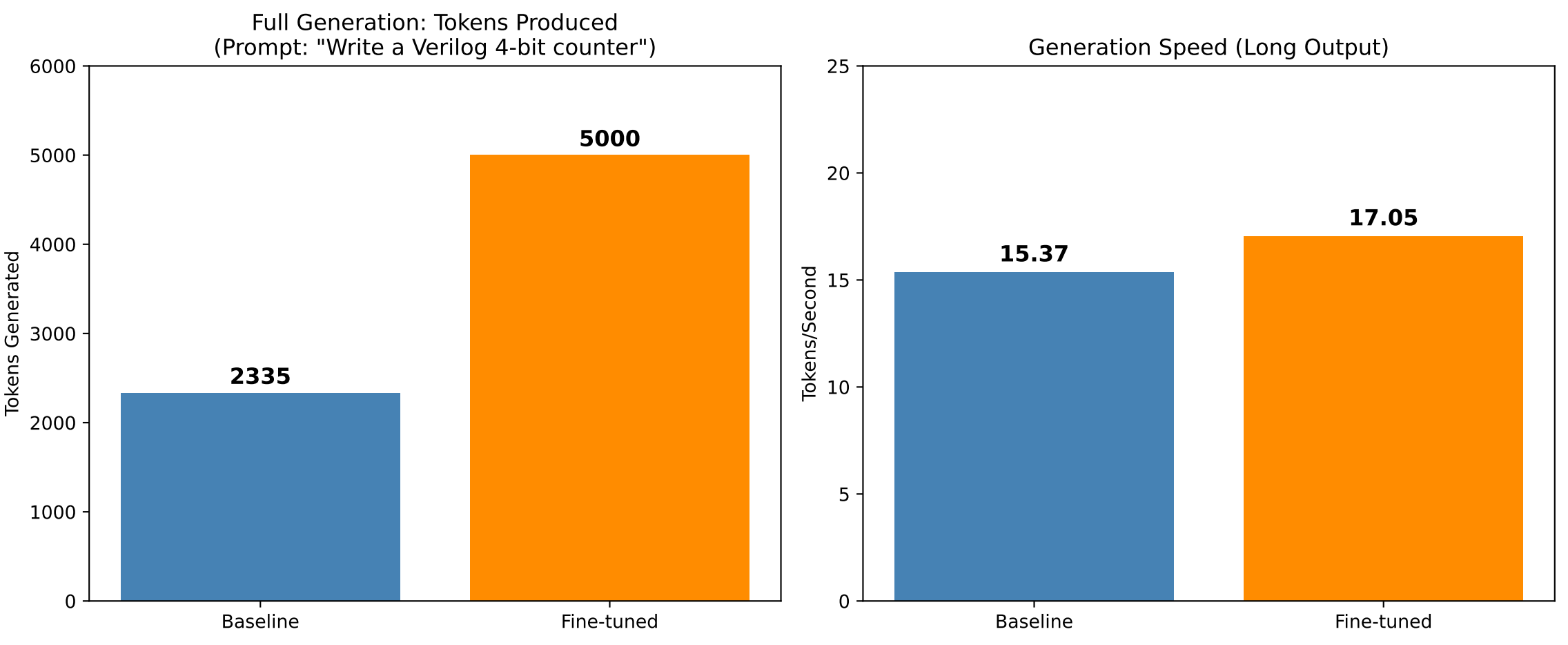
   - Long generations (2000+ tokens):
     * Baseline: 15.37 tok/s (completes with code)
     * Fine-tuned: 17.05 tok/s (gets stuck in reasoning)

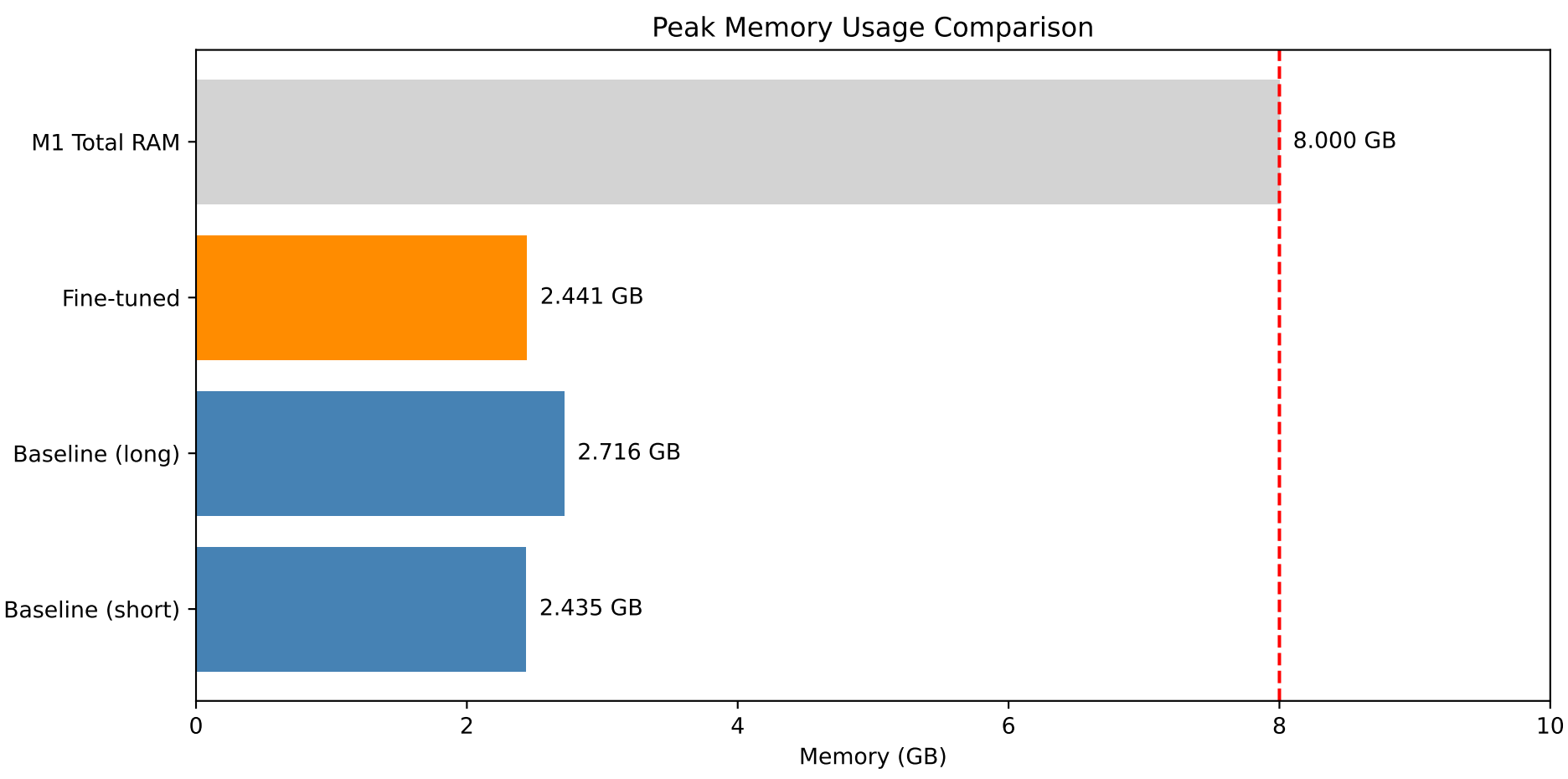
2. MEMORY USAGE:
   - Baseline peak: 2.435 - 2.716 GB
   - Fine-tuned peak: 2.441 GB
   - Both models fit comfortably in 8GB M1

3. OUTPUT QUALITY:
   - Baseline: Produces complete, working Verilog code
   - Fine-tuned: Extended reasoning, but may not complete code

4. RECOMMENDATION:
   For EDA/Verilog tasks, the baseline model provides
   better practical results. The fine-tuned model shows
   extensive reasoning but needs output limiting.


Generation Speed (500 token limit)
Tokens/Second
Baseline
Fine-tuned
0
20
40
60
80
100
120
4-bit counter
D flip-flop
2-to-1 MUX
Traffic FSM
4-bit adder TB
Response Time (500 token limit)
Time (seconds)
Baseline
Fine-tuned
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
4-bit counter
D flip-flop
2-to-1 MUX
Traffic FSM
4-bit adder TB


Full Generation: Tokens Produced
(Prompt: "Write a Verilog 4-bit counter")
Tokens Generated
6000
5000
4000
3000
2000
1000
0
2335
5000
Baseline
Fine-tuned
Generation Speed (Long Output)
Tokens/Second
25
20
15
10
5
0
15.37
17.05
Baseline
Fine-tuned


Peak Memory Usage Comparison
M1 Total RAM
Fine-tuned
Baseline (long)
Baseline (short)
8.000 GB
2.441 GB
2.716 GB
2.435 GB
0
2
4
6
8
10
Memory (GB)

DETAILED BENCHMARK RESULTS
==========================

SHORT GENERATION TESTS (500 token limit):

Prompt	Baseline TPS	Fine-tuned TPS	Baseline Time	Fine-tuned Time
4-bit counter	58.19	99.63	1.39s	0.81s
D flip-flop	99.54	100.46	0.81s	0.81s
2-to-1 MUX	101.43	99.91	0.80s	0.81s
Traffic FSM	101.51	98.81	0.80s	0.82s
4-bit adder TB	97.80	90.14	0.83s	0.90s
AVERAGE	91.70	97.79	0.93s	0.83s

FULL GENERATION TEST (2M token limit):

Metric	Baseline	Fine-tuned
Tokens Generated	2,335	5,000+ (killed)
Prompt Eval (tok/s)	20.61	21.51
Generation (tok/s)	15.37	17.05
Peak Memory (GB)	2.716	2.441
Code Complete?	YES	NO (loop)
Code Quality	Excellent	Incomplete

NOTES:
- Fine-tuned model shows extended reasoning (thinking mode)
- Baseline produces more concise, actionable output
- Both models perform well within M1 8GB memory constraints

OUTPUT QUALITY ANALYSIS
=======================

BASELINE MODEL OUTPUT (4-bit counter prompt):

- Generated complete, working Verilog module
- Included proper module declaration with ports
- Added synchronous reset (active low)
- Provided clear comments and documentation
- Generated simulation behavior table
- Total: 2,335 tokens with complete code

Example output structure:
  * Module declaration
  * Register definition
  * Always block with clock and reset
  * Complete endmodule
  * Usage notes and testbench hints

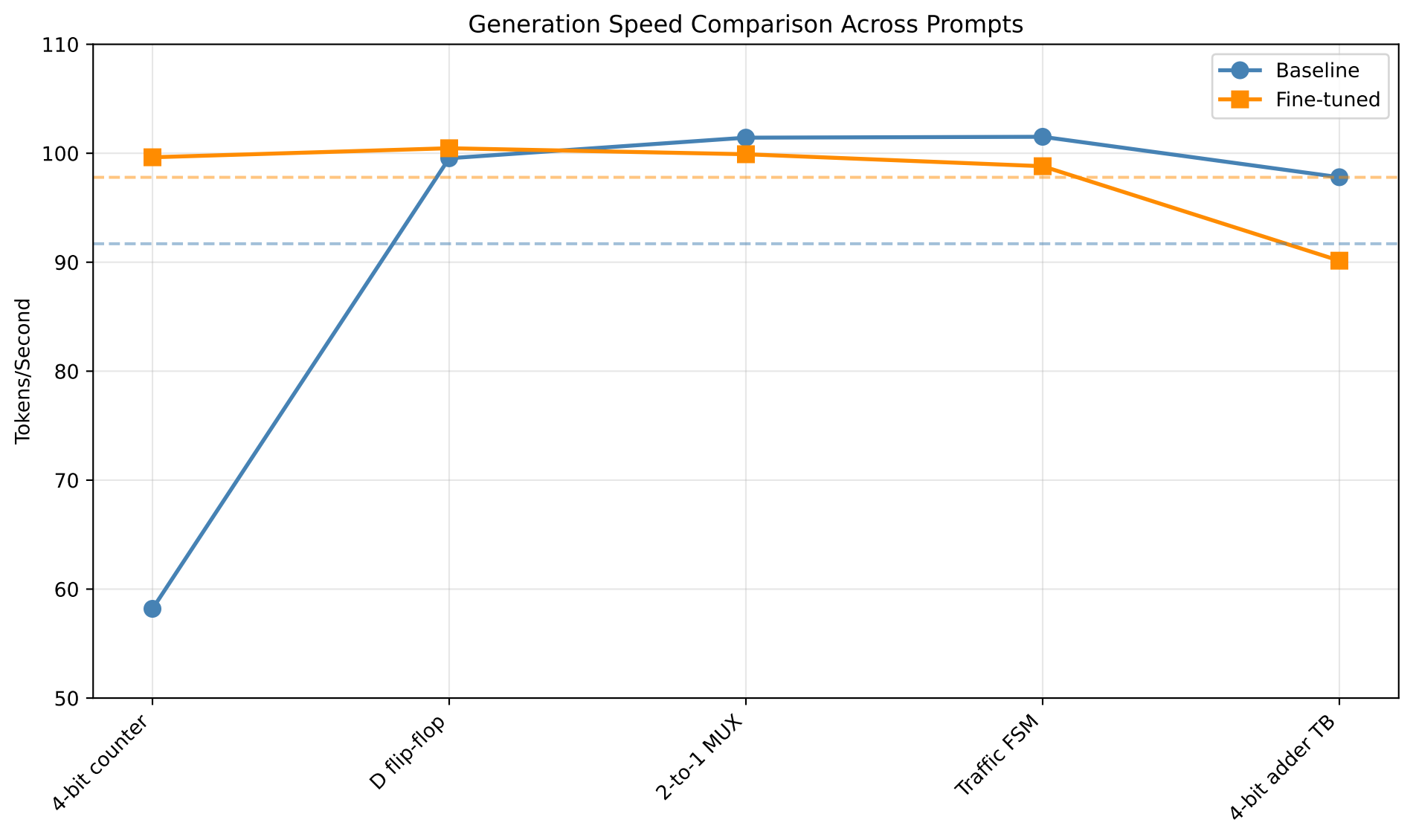
FINE-TUNED MODEL OUTPUT (4-bit counter prompt):

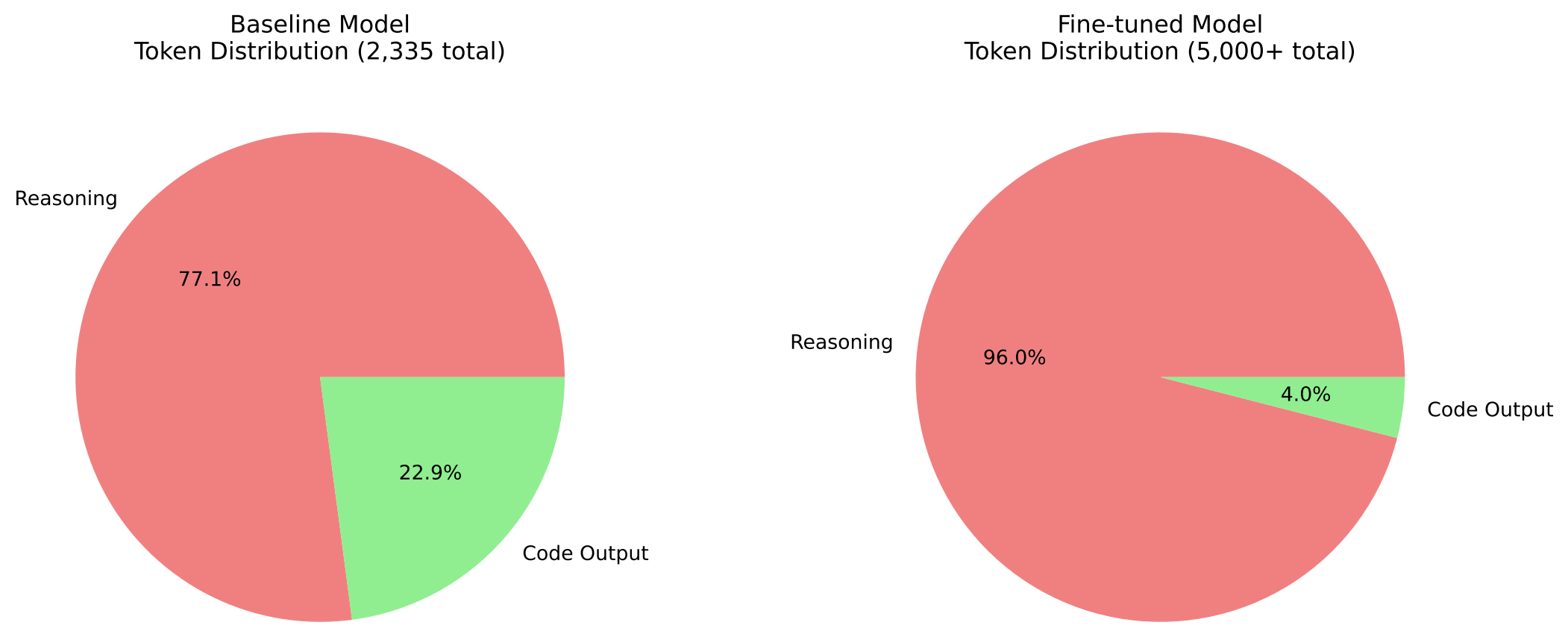
- Generated extensive theoretical analysis
- Discussed multiple counter design approaches
- Explored synchronous vs asynchronous design
- Got stuck in reasoning loop about next-state logic
- Did not complete the actual Verilog implementation
- Total: 5,000+ tokens (still generating when killed)

Characteristics observed:
  * Very detailed thinking process
  * Multiple design considerations explored
  * Extensive XOR/AND logic derivation attempts
  * Repeated reasoning patterns (possible loop)
  * Incomplete final implementation

CONCLUSION:

The baseline model is more suitable for practical code generation tasks.
The fine-tuned model shows signs of over-thinking which may require:
  1. Lower max_tokens limit
  2. Temperature adjustment
  3. Different fine-tuning approach for reasoning models
  4. System prompt to encourage concise output


Generation Speed Comparison Across Prompts
Baseline
Fine-tuned
Tokens/Second
110
100
90
80
70
60
50
4-bit counter
D flip-flop
2-to-1 MUX
Traffic FSM
4-bit adder TB


Baseline Model
Token Distribution (2,335 total)
Reasoning
77.1%
22.9%
Code Output
Fine-tuned Model
Token Distribution (5,000+ total)
Reasoning
96.0%
4.0%
Code Output

RECOMMENDATIONS & NEXT STEPS
============================

CURRENT STATE ANALYSIS:
----
1. Baseline Model: Ready for production use
   - Generates complete, working code
   - Good balance of reasoning and output
   - Consistent performance across prompts

2. Fine-tuned Model: Needs optimization
   - Shows excessive reasoning (thinking mode issue)
   - May need training data adjustment
   - Consider limiting reasoning tokens

RECOMMENDED OPTIMIZATIONS:
----
1. For Fine-tuned Model:
   - Add </think> token training to end reasoning
   - Use max_tokens limit of 1000-2000 for practical use
   - Consider adding system prompt: "Be concise, show code first"
   - Review training data for proper response structure

2. For Production Use:
   - Use baseline model for reliable code generation
   - Use fine-tuned model for educational/explanation tasks
   - Implement token streaming for better UX

3. Memory Optimization:
   - Both models fit well in 8GB M1
   - Consider 2-bit quantization for smaller models
   - KV cache compression could help with longer contexts

HARDWARE CONSIDERATIONS (M1 8GB):

- Current memory usage: ~2.5GB (very comfortable)
- Room for larger context windows
- Could run larger models (8B) with optimization
- Metal GPU acceleration working properly

BENCHMARK METHODOLOGY IMPROVEMENTS:
----
1. Add warm-up runs to stabilize timings
2. Use multiple iterations per prompt
3. Add code correctness validation (syntax check)
4. Include perplexity measurements
5. Test with varied temperatures

Report generated: 2025-12-09 00:35:00