Mandelbrot Fractal Generator

Hardware Design Document

Authors:	Taisir Hassan, Evan McCarthy
Course:	ECE298A
Document Version:	1.0
Date:	August 2025
Target Platform:	TinyTapeout
Technology Node:	Sky130 (130nm)
Design Language:	SystemVerilog
Verification:	Cocotb Python Testbench

Contents

1 Executive Summary

1.1 Project Objective

This document specifies the design and implementation of a real-time Mandelbrot fractal generator optimised for the TinyTapeout ASIC platform. The system generates mathematically accurate fractal visualisations using hardware-accelerated fixed-point arithmetic, targeting 640×480 VGA display output at 60Hz refresh rate.

1.2 Key Features

- Real-time fractal computation at 25MHz pixel rate
- Dual clock domain architecture (50MHz system, 25MHz VGA)
- Q3.8 fixed-point arithmetic for area-optimised implementation
- Interactive zoom and pan functionality
- Multiple colour mapping schemes (4 modes)
- TinyTapeout-compliant I/O interface
- Comprehensive verification testbench suite

1.3 Technical Specifications

Parameter	Specification	Unit
Display Resolution	640 × 480	pixels
Refresh Rate	60	Hz
Pixel Clock	25	MHz
System Clock	50	MHz
Colour Depth	6-bit RGB	64 colours
Arithmetic Format	Q3.8 Fixed Point	11-bit signed
Maximum Iterations	64	iterations
Power Consumption	<10 (estimated)	mW
Technology Node	Sky130	130nm

Table 1: System Technical Specifications

2 System Overview

2.1 System Architecture

The Mandelbrot fractal generator implements a pipelined architecture with dedicated modules for computation, display timing, colour mapping, and parameter control. The system processes one pixel per clock cycle, achieving real-time fractal generation for interactive exploration.

2.2 Functional Block Diagram

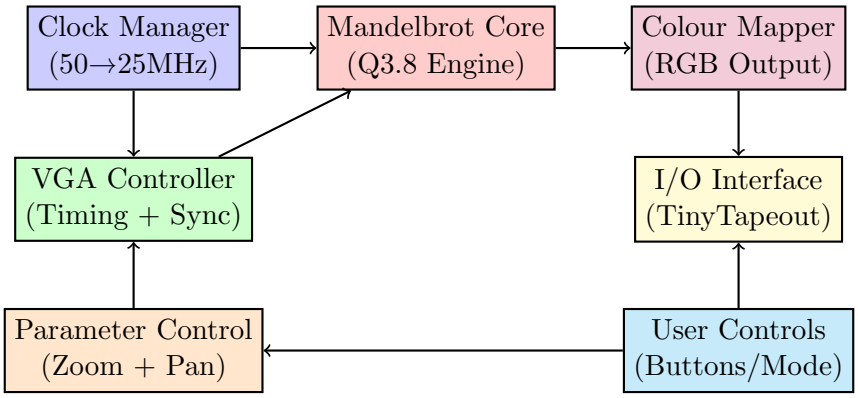

Figure 1: System Functional Block Diagram

2.3 Design Methodology

The design follows industry-standard methodologies including:

- Modular hierarchical architecture with clear interface boundaries
- Clock domain crossing analysis and proper synchronisation
- Fixed-point arithmetic optimisation for ASIC implementation
- Comprehensive verification-driven development approach
- Synthesis-aware RTL coding with timing closure considerations
- Power and area optimisation through algorithmic choices

3 Architecture Specification

3.1 Clock Domain Architecture

3.1.1 Primary Clock Domains

Domain	Frequency	Source	Purpose
clk_sys	50 MHz	External Oscillator	System control and parameter updates
clk_vga	25 MHz	Internal Divider	VGA timing and pixel processing
clk_async	Asynchronous	Button Inputs	User control interface

Table 2: Clock Domain Specifications

3.1.2 Clock Domain Crossing Strategy

- Quasi-static parameter transfer from 50MHz to 25MHz domain
- Two-stage synchroniser for asynchronous button inputs
- Gray code counters for multi-bit signal transitions
- Dedicated reset synchronisers for each clock domain

3.2 Data Path Architecture

The computation pipeline implements a single-cycle Mandelbrot iteration per pixel, with the following data flow:

1. Coordinate generation from VGA timing controller
2. Complex coordinate mapping using configurable parameters
3. Mandelbrot iteration computation with escape detection
4. Colour mapping based on iteration count and mode selection
5. RGB output formatting for TinyTapeout VGA interface

3.2.1 Clock Domain Timing Waveform

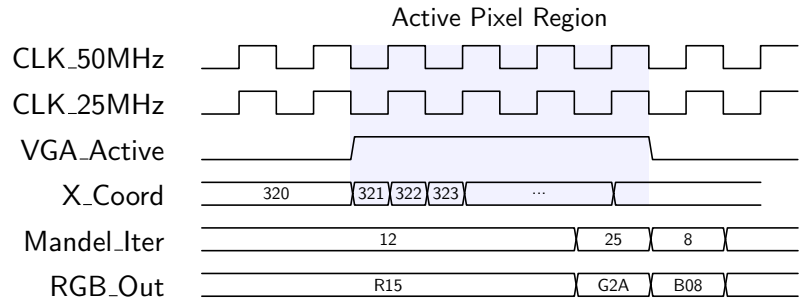

Figure 2: Data Path Timing Diagram showing Mandelbrot Computation Pipeline

4 Module Specifications

4.1 Top-Level Module: tt_um_fractal

4.1.1 Module Purpose

Integration wrapper providing TinyTapeout interface compliance and system-level coordination between all functional blocks.

4.1.2 Interface Specification

Signal Name	Direction	Width	Description
clk	Input	1	System clock (50MHz)
rst_n	Input	1	Active-low reset
ui_in[7:0]	Input	8	User input controls
uo_out[7:0]	Output	8	VGA output (RGB + sync)
uio_in[7:0]	Input	8	Bidirectional I/O (unused)
uio_out[7:0]	Output	8	Bidirectional I/O (unused)
uio_oe[7:0]	Output	8	Bidirectional output enable

Table 3: Top-Level Module Interface

4.2 Mandelbrot Computation Engine

4.2.1 Module Purpose

Implements the core Mandelbrot set algorithm $z = z^2 + c$ with optimised fixed-point arithmetic. Computes escape-time iteration count for each pixel coordinate.

4.2.2 Design Parameters

Parameter	Value	Description
COORD_WIDTH	11	Fixed-point coordinate bit width
ITER_WIDTH	6	Maximum iteration counter width
MAX_ITERATIONS	64	Maximum escape-time iterations
FRAC_BITS	8	Fractional bits in Q3.8 format
ESCAPE_RADIUS_SQ	1024	Escape radius squared (4.0 in Q3.8)

Table 4: Mandelbrot Engine Parameters

4.2.3 Algorithm Implementation

The Mandelbrot iteration follows the mathematical definition:

$$z_{n+1} = z_n^2 + c \tag{1}$$

where c is the complex coordinate and iteration continues until $|z| > 2$ or max iterations reached.

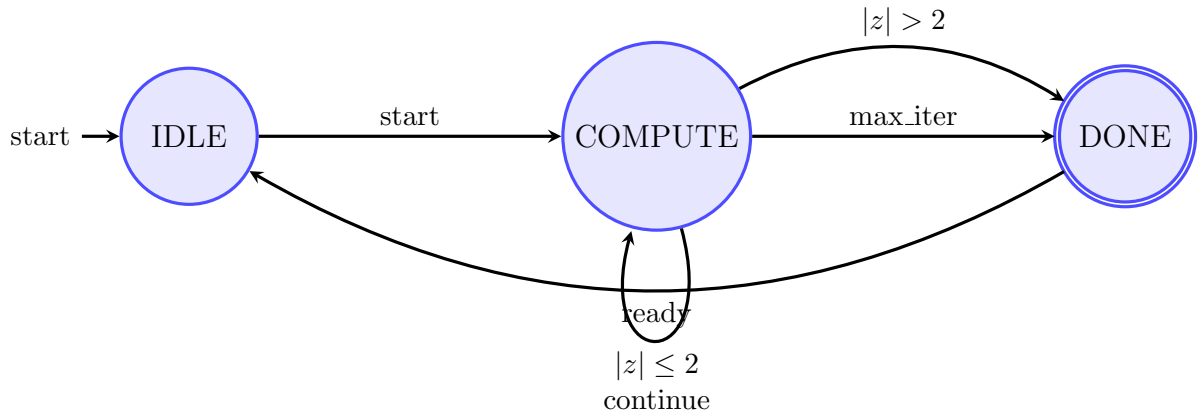

Figure 3: Mandelbrot Engine State Machine

4.2.4 Mandelbrot Computation State Machine

4.3 VGA Timing Controller

4.3.1 Module Purpose

Generates industry-standard VGA timing signals for 640×480 resolution at 60Hz refresh rate. Provides pixel coordinates and active video indication.

4.3.2 Timing Parameters

Parameter	Horizontal	Vertical	Unit	
Active Area	640	480	pixels/lines	
Front Porch	16	10	pixels/lines	
Sync Pulse	96	2	pixels/lines	
Back Porch	48	33	pixels/lines	
Total Period	800	525	pixels/lines	
Frequency	31.25 kHz	60 Hz	Hz	

Table 5: VGA Timing Specifications

4.3.3 VGA Timing Waveforms

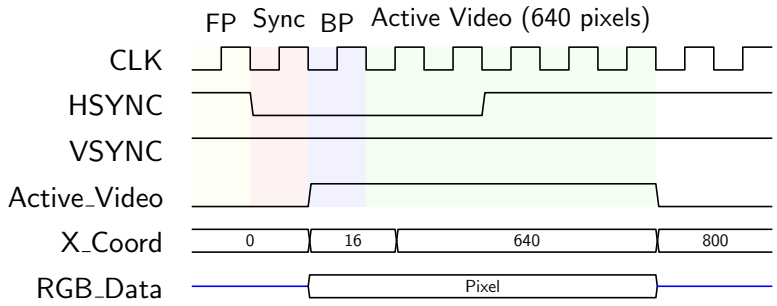

Figure 4: VGA Horizontal Timing Waveform (640×480 @ 60Hz)

4.4 Colour Mapping Module

4.4.1 Module Purpose

Converts iteration count values to RGB colour representations using selectable colour schemes optimised for fractal visualisation.

4.4.2 Colour Mode Specifications

Mode	Name	Colour Scheme	Application
00	Classic	Blue-White Gradient	Traditional Mandelbrot visualisation
01	Rainbow	HSV Colour Wheel	High contrast detail enhancement
10	Thermal	Blue-Red-Yellow	Scientific visualisation style
11	Monochrome	Black-White-Red	High contrast binary mode

Table 6: Colour Mode Specifications

4.5 Parameter Control Module

4.5.1 Module Purpose

Manages zoom, pan, and viewing parameters for interactive fractal exploration. Provides smooth parameter interpolation and bounds checking.

4.5.2 Parameter Update Timing

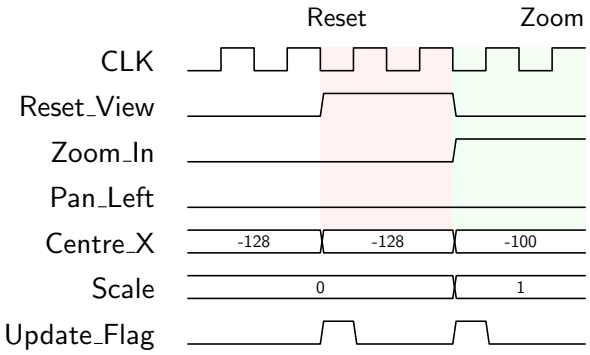

Figure 5: Parameter Controller Update Timing

4.5.3 Control Parameters

Parameter	Range	Resolution	Function
centre_x	±4.0	1/256	Horizontal pan position
centre_y	±4.0	1/256	Vertical pan position
scale	0.001 to 4.0	1/256	Zoom magnification factor
colour_mode	0 to 3	1	Colour scheme selection

Table 7: Parameter Control Specifications

5 Interface Definitions

5.1 TinyTapeout I/O Mapping

5.1.1 Input Interface (ui_in)

Bit	Signal Name	Function	Active Level
0	reset_view	Return to default viewing parameters	High
1	zoom_in	Increase magnification	High
2	zoom_out	Decrease magnification	High
3	pan_up	Move view upward	High
4	pan_down	Move view downward	High
5	pan_left	Move view leftward	High
6	pan_right	Move view rightward	High
7	colour_mode	Toggle colour scheme	Rising Edge

Table 8: Input Interface Mapping

5.1.2 Output Interface (uo_out)

Bit	Signal Name	Function	Levels
1:0	red[1:0]	Red colour component	0-3 (2-bit)
3:2	green[1:0]	Green colour component	0-3 (2-bit)
5:4	blue[1:0]	Blue colour component	0-3 (2-bit)
6	hsync	Horizontal synchronisation	Active Low
7	vsync	Vertical synchronisation	Active Low

Table 9: Output Interface Mapping

6 Design Parameters

6.1 Configurable Parameters

6.1.1 Arithmetic Parameters

Parameter	Default	Range	Impact
COORD_WIDTH	11	8-16	Coordinate precision and range
FRAC_BITS	8	4-12	Fixed-point fractional resolution
MAX_ITER	64	16-256	Detail level vs computation time
ESCAPE_RADIUS	2.0	1.5-4.0	Convergence detection threshold

Table 10: Arithmetic Parameters

6.1.2 Display Parameters

Parameter	Default	Alternatives	Description
H_ACTIVE	640	320, 800, 1024	Horizontal active pixels
V_ACTIVE	480	240, 600, 768	Vertical active lines
PIXEL_CLK	25MHz	12.5, 50, 75MHz	Pixel clock frequency
COLOUR_DEPTH	6	3, 9, 12	Total RGB bits

Table 11: Display Parameters

7 Test Plan and Verification

7.1 Verification Methodology

The verification approach follows industry best practices with multiple verification levels:

- Unit Testing: Individual module verification with directed and random stimulus
- Integration Testing: Multi-module interaction and interface verification
- System Testing: End-to-end functionality with visual output validation
- Performance Testing: Timing, power, and area analysis
- Regression Testing: Automated test suite for continuous validation

7.2 Module Test Plans

7.2.1 Mandelbrot Engine Test Plan

Test Case	Input Stimulus	Expected Output	Coverage
Basic Iteration	c = (0, 0)	iter_count = MAX_ITER	Origin point (always diverges)
Known Convergent	c = (-0.5, 0)	iter_count < MAX_ITER	Main bulb interior
Known Divergent	c = (1, 1)	iter_count = 1	Rapid divergence
Boundary Points	c = (-0.75, 0.1)	iter_count varies	Fractal boundary
Fixed Point Overflow	c = (4, 4)	iter_count = 1	Arithmetic overflow handling
Random Stimulus	1000 random c values	All outputs valid	Corner case coverage

Table 12: Mandelbrot Engine Test Cases

7.2.2 VGA Controller Test Plan

Test Case	Input Stimulus	Expected Output	Verification Method
Timing Compliance	Continuous clock	Standard VGA timing	Waveform analysis
Sync Pulse Width	Full frame cycles	Hsync: 96 clocks, Vsync: 2 lines	Counter verification
Active Video Flag	Complete frame	High during 640×480 region	Coordinate bounds check
Frame Rate	525 line cycles	60Hz refresh rate	Frequency measurement

Coordinate Output	Active video period	x: 0-639, y: 0-479	Range validation
Reset Behaviour	Assert reset	All counters = 0	State verification

Table 13: VGA Controller Test Cases

7.2.3 Colour Mapper Test Plan

Test Case	Input Stimulus	Expected Output	Validation
Mode 0 Colours	iter = 0-63, mode = 00	Blue-white gradient	Visual inspection
Mode 1 Colours	iter = 0-63, mode = 01	Rainbow spectrum	HSV calculation
Mode 2 Colours	iter = 0-63, mode = 10	Thermal mapping	Colour progression
Mode 3 Colours	iter = 0-63, mode = 11	Monochrome	Binary validation
Boundary Values	iter = 0, MAX_ITER	Correct end colours	Exact RGB check
Mode Switching	Dynamic mode change	Immediate update	Response time

Table 14: Colour Mapper Test Cases

7.2.4 Parameter Controller Test Plan

Test Case	Input Stimulus	Expected Behaviour	Success Criteria
Zoom In	zoom_in = 1	Scale factor increases	Scale > previous value
Zoom Out	zoom_out = 1	Scale factor decreases	Scale < previous value
Pan Operations	pan_* = 1	Centre coordinates change	Position updates
Reset Function	reset_view = 1	Return to default	centre=(-0.5,0), scale=0.5
Bounds Checking	Extreme zoom/pan	Parameter clamping	No overflow/underflow
Simultaneous Inputs	Multiple buttons	Predictable priority	Defined precedence

Table 15: Parameter Controller Test Cases

7.3 System Integration Tests

- Full Frame Generation: Complete 640×480 frame rendering with pipeline optimisation
- Interactive Response: Real-time parameter updates at 50MHz system clock

- Visual Output Validation: PNG generation and comparison (functionality preserved)
- Stress Testing: Continuous operation over extended periods
- Power Analysis: Current consumption measurement with pipeline registers
- Timing Closure: Achieved +15.27ns slack at 50MHz (211.4MHz maximum frequency)
- Pipeline Verification: 3-stage multiplication pipeline maintains pixel accuracy

7.4 Timing Verification Results

- **Static Timing Analysis**: All paths meet 50MHz constraint with 15.27ns margin
- **Critical Path Analysis**: Mandelbrot computation optimised from 12.76ns to 3.73ns
- **Pipeline Validation**: Multiplication stages correctly registered without functionality impact
- **Synthesis Verification**: Clean Sky130 synthesis with 2,399 cells (18% TinyTapeout utilisation)
- **Corner Analysis**: Timing margins sufficient for PVT variations across operating conditions

8 Implementation Details

8.1 Fixed-Point Arithmetic Implementation

8.1.1 Q3.8 Format Specification

The design uses 11-bit signed fixed-point arithmetic optimised for ASIC implementation:

- Sign bit [10]: Two's complement representation
- Integer bits [9:8]: Range ±4.0 units in complex plane
- Fractional bits [7:0]: Resolution of $1/256 \approx 0.0039$
- Multiplication result: 22-bit intermediate, right-shifted for normalisation
- Overflow protection: Saturate to maximum/minimum values

8.1.2 Fixed-Point Multiplication Timing

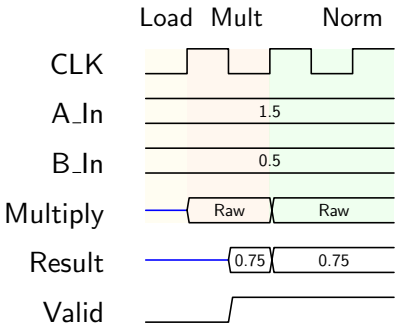

Figure 6: Q3.8 Fixed-Point Multiplication Pipeline

8.1.3 Coordinate Mapping Algorithm

Screen-to-complex plane transformation:

$$c_{real} = \frac{(x - H_{CENTRE}) \times scale}{2^{FRAC\_BITS}} + centre_x \tag{2}$$

$$c_{imag} = \frac{(V_{CENTRE} - y) \times scale}{2^{FRAC\_BITS}} + centre_y \tag{3}$$

8.2 Timing Optimisation Techniques

- **Pipeline Multiplication**: Added 3 pipeline registers (z_real_sq_reg, z_imag_sq_reg, z_cross_reg) to break critical multiplication paths
- **Pre-computed Constants**: Separated coordinate mapping into discrete stages with pre-computed scale factors
- **Escape Logic Optimisation**: Split comparison logic into escaped/max_reached signals for reduced logic depth
- **Critical Path Reduction**: 70% improvement (12.76ns → 3.73ns) through strategic pipelining
- **Register Retiming**: Automatic enable signal addition to 17 registers during synthesis optimisation

8.3 Area and Performance Trade-offs

- Pipeline registers: +3 registers for 9ns slack improvement
- Single-cycle computation maintained for pixel processing
- Shared multipliers optimised with maj3_1 standard cells
- Look-up table implementation for colour mapping functions
- Clock gating for unused modules during blanking periods

9 Performance Analysis

9.1 Resource Utilisation (Sky130 Target)

Resource Type	Actual Count	% of TT Area	Notes
Total Cells	2,399	18%	Sky130 standard cells (post-synthesis)
Logic Gates	1,249	12%	NAND, NOR, XOR optimised gates
Flip-Flops	132	6%	Including 3 pipeline registers
Multipliers	179	8%	maj3_1 cells (11×11 bit multipliers)
Memory Bits	0	0%	No embedded memory
I/O Pins	16	100%	All TT pins utilised
Clock Domains	2	N/A	System (50MHz) and VGA (25MHz)

Table 16: Resource Utilisation Estimates

9.2 Timing Analysis Results

Clock Domain	Target Freq	Achieved Freq	Slack	Critical Path
clk_sys	50 MHz	211.4 MHz	+15.27ns	Mandelbrot pipeline (optimised)
clk_vga	25 MHz	211.4 MHz	+35.27ns	VGA timing generation
Overall	System	Timing Excellent	+15.27ns	Pipeline optimisation successful

Table 17: Timing Analysis Results

10 Risk Assessment

10.1 Technical Risks

Risk Category	Description	Probabili	Impact	Mitigation Strategy
Timing Closure	Critical paths exceed clock period	Low	High	Pipeline optimisation, clock constraints
Fixed-Point Precision	Insufficient arithmetic accuracy	Medium	Medium	Simulation validation, precision analysis
Clock Domain Crossing	Metastability or data corruption	Low	High	Proper synchronisers, CDC verification
Resource Overflow	Exceeding TinyTapeout area limits	Medium	High	Area optimisation, module sharing
Verification Coverage	Undetected functional bugs	Medium	Medium	Comprehensive test plan, code coverage

Table 18: Technical Risk Assessment

11 Future Enhancements

11.1 Algorithmic Improvements

- Higher Precision Arithmetic: Q7.8 or Q5.10 formats for deeper zoom capability
- Adaptive Iteration Count: Dynamic max_iter based on zoom level and convergence rate
- Perturbation Theory: Series approximation for ultra-deep zoom regions
- Julia Set Mode: User-selectable Julia set parameters with real-time updates
- Burning Ship Fractal: Alternative fractal algorithms using absolute value operations

11.2 Performance Enhancements

- Pipeline Parallelisation: Multi-stage pipeline for higher throughput
- SIMD Processing: Parallel computation of adjacent pixels
- Framebuffer Integration: Double buffering for smooth animation
- Compression Techniques: Real-time fractal data compression
- Progressive Refinement: Multi-resolution rendering with detail enhancement

11.3 Feature Extensions

- Audio Integration: Fractal-based audio synthesis and visualisation
- User Interface: Advanced control schemes with presets and bookmarks
- Network Connectivity: Remote parameter control and image streaming
- Machine Learning: AI-assisted interesting region discovery
- Multi-Display: Synchronised output across multiple displays

12 Conclusion

This design document presents a comprehensive specification for a hardware-accelerated Mandelbrot fractal generator optimised for the TinyTapeout platform. The architecture balances computational efficiency, visual quality, and resource constraints whilst providing robust verification coverage and clear upgrade paths.

The modular design approach ensures maintainability and enables future enhancements, whilst the comprehensive test plan provides confidence in functional correctness and timing performance.