AMBA Bus Matrix Configuration Tool

Visual Guide with Conceptual Mockups

⚠ Contains Mockups, Not Real Screenshots ⚠

Version 1.0.0

July 2025

⚠ IMPORTANT DISCLAIMER ⚠

CURRENT STATUS OF GUI IMAGES:

□ NOT REAL SCREENSHOTS
The images in this guide are PROGRAMMATICALLY GENERATED MOCKUPS created with matplotlib, not actual screenshots from the running GUI application.

□ WHAT THESE IMAGES ARE:
- Conceptual representations of GUI layout and workflow
- Based on analysis of the actual GUI source code (bus_matrix_gui.py)
- Designed to show expected interface structure and functionality
- Created to demonstrate the intended user workflow

□ WHAT THESE IMAGES ARE NOT:
- Actual screenshots from a running GUI application
- Real captures of the interface in action
- Guaranteed to match the exact appearance of the running application

□ TO GET REAL SCREENSHOTS:
1. Launch the actual GUI: ./launch_gui.sh
2. Follow the workflow steps described in this guide
3. Use screenshot tools (Print Screen, snipping tool, etc.)
4. Replace mockup images with real captures

□ VERIFICATION NEEDED:
This documentation needs to be updated with real screenshots from the actual running GUI application to provide authentic visual guidance.

□ PURPOSE OF THIS GUIDE:
Until real screenshots are available, this guide provides the conceptual workflow and expected interface behavior based on the source code analysis.

Mockup vs Real GUI Comparison

CURRENT: Programmatic Mockups

• Created with matplotlib rectangles/text
• Based on GUI source code analysis
• Shows expected layout structure
• Colors match actual GUI constants:
  - Master: #4CAF50 (green)
  - Slave: #2196F3 (blue)
  - Interconnect: #FFC107 (amber)
• Demonstrates workflow concepts
• NOT captured from running application

NEEDED: Real GUI Screenshots

• Captured from running bus_matrix_gui.py
• Shows actual Tkinter interface
• Real drag-and-drop interactions
• Authentic dialog boxes
• Actual button/menu appearances
• True canvas rendering
• Live configuration panels
• Real-time validation messages

How to Replace with Real Screenshots

1. Navigate to: /home/timtim01/eda_test/project/gen_amba_2025/axi4_vip/gui/
2. Run: ./launch_gui.sh (requires GUI environment with DISPLAY)
3. Take screenshots at each workflow step:
   - Main window after launch
   - Add Master dialog (click "Add Master" button)
   - Canvas with 2 masters + 3 slaves designed
   - RTL Generation dialog (click "Generate RTL")
   - File browser showing generated outputs
   - VIP Generation process (click "Generate VIP")
4. Save as PNG files with same names as current mockups
5. Regenerate this PDF with real images embedded

GUI Source Code Analysis

ANALYSIS OF bus_matrix_gui.py (Actual GUI Application):

□□ GUI FRAMEWORK:
- Built with Python Tkinter
- Uses tk.Canvas for visual bus matrix representation
- Implements drag-and-drop functionality for masters/slaves
- Has zoom and grid snapping capabilities

□ VISUAL ELEMENTS:
- Masters: Green blocks (#4CAF50) with drag handles
- Slaves: Blue blocks (#2196F3) with address/size info
- Interconnect: Amber color (#FFC107) for connections
- Connection lines: Gray (#757575) with arrows
- Selected items flash with yellow highlight
- Security indicators: [SEC] and [PERM] tags

□ GUI COMPONENTS:
- BusMatrixCanvas - main visual design area
- Master/Slave configuration dialogs
- Properties panels for component configuration
- Menu bar with File, Edit, View, Tools, Generate options
- Toolbar with quick access buttons
- Status bar for validation messages

⚙□ FUNCTIONALITY:
- Add/remove masters and slaves visually
- Drag to reposition components on canvas
- Right-click context menus for configuration
- Address overlap detection and validation
- RTL generation with progress dialogs
- VIP generation with UVM output
- Configuration save/load (JSON format)
- Integration with AXI Verilog generator

□ CONFIGURATION CLASSES:
- MasterConfig: name, id_width, qos_support, priority, etc.
- SlaveConfig: name, base_address, size, memory_type, etc.
- BusConfig: bus_type, data_width, addr_width, arbitration

□ KEY FEATURES IDENTIFIED:
- Visual bus matrix design with real-time validation
- Support for AXI4, AXI3, AHB, APB protocols
- Security and permission controls
- QoS and priority configuration
- Export to synthesizable Verilog RTL
- Complete UVM verification environment generation

This analysis confirms the GUI is a fully functional application that can be launched and used to create real screenshots for documentation.

Expected GUI Workflow (Conceptual)

⚠ CONCEPTUAL MOCKUPS BELOW - NOT REAL SCREENSHOTS ⚠

```
STEP-BY-STEP WORKFLOW (Based on Source Code Analysis):

1️⃣ LAUNCH GUI
   Command: ./launch_gui.sh
   Expected: Tkinter window opens with canvas, toolbar, properties panel
   [MOCKUP PLACEHOLDER - Need real screenshot of main window]

2️⃣ ADD MASTER
   Action: Click "Add Master" button in toolbar
   Expected: Dialog box with fields for Name, ID Width, Priority, QoS
   [MOCKUP PLACEHOLDER - Need real screenshot of add master dialog]

3️⃣ ADD SLAVES
   Action: Click "Add Slave" button, configure addresses
   Expected: Dialog with Name, Base Address, Size, Memory Type fields
   [MOCKUP PLACEHOLDER - Need real screenshot of add slave dialog]

4️⃣ DESIGN CANVAS
   Result: Canvas shows green master blocks, blue slave blocks
   Expected: Drag-and-drop repositioning, connection lines, address labels
   [MOCKUP PLACEHOLDER - Need real screenshot of design canvas]

5️⃣ VALIDATE DESIGN
   Action: Tools → Validate Design
   Expected: Status messages, address overlap checking
   [MOCKUP PLACEHOLDER - Need real screenshot of validation]

6️⃣ GENERATE RTL
   Action: Generate → Generate RTL
   Expected: Dialog showing file list, output directory, progress bar
   [MOCKUP PLACEHOLDER - Need real screenshot of RTL generation]

7️⃣ GENERATE VIP
   Action: Generate → Generate VIP
   Expected: VIP generation dialog, UVM environment creation
   [MOCKUP PLACEHOLDER - Need real screenshot of VIP generation]

8️⃣ VIEW OUTPUT
   Result: File browser showing generated .v files and VIP structure
   [MOCKUP PLACEHOLDER - Need real screenshot of output files]

ACCURACY NOTE:
This workflow is based on source code analysis of bus_matrix_gui.py.
The actual GUI may have additional features, different layouts, or
modified behavior that can only be captured through real screenshots.
```