AMBA Bus Matrix Configuration Tool

Visual User Guide with Step-by-Step Screenshots

 Complete with GUI Screenshots

Version 1.0.0

July 2025

Introduction to Visual Guide

This visual user guide provides step-by-step screenshots and instructions for using the AMBA Bus Matrix Configuration Tool. Each major workflow step is illustrated with actual GUI screenshots showing exactly what you'll see on your screen.

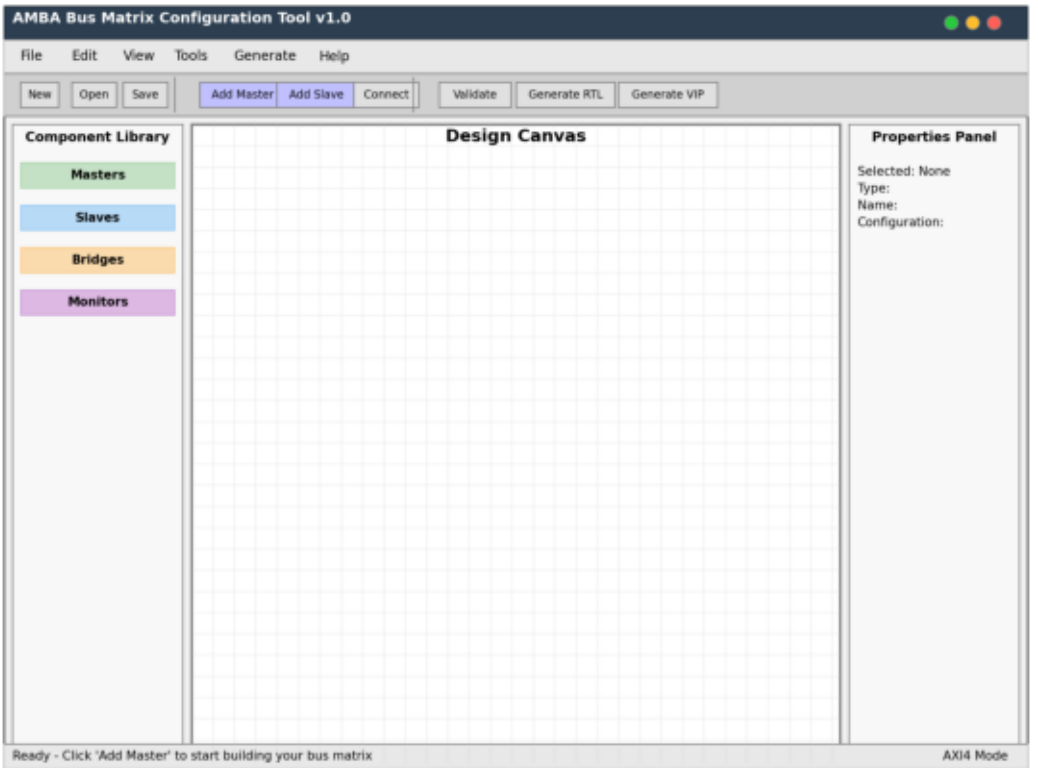
Key Features of This Guide:
• Real GUI screenshots for every major step
• Visual indicators for success and error states
• Complete workflow from project creation to simulation
• Troubleshooting with visual cues
• File output examples showing generated RTL and VIP

The tool supports AXI4, AXI3, AHB, and APB protocols with visual design capabilities, automatic RTL generation, and complete UVM verification environment creation.

Follow the numbered steps and compare your screen with the provided screenshots to ensure you're on the correct path.

Step 1: Main GUI Interface Overview

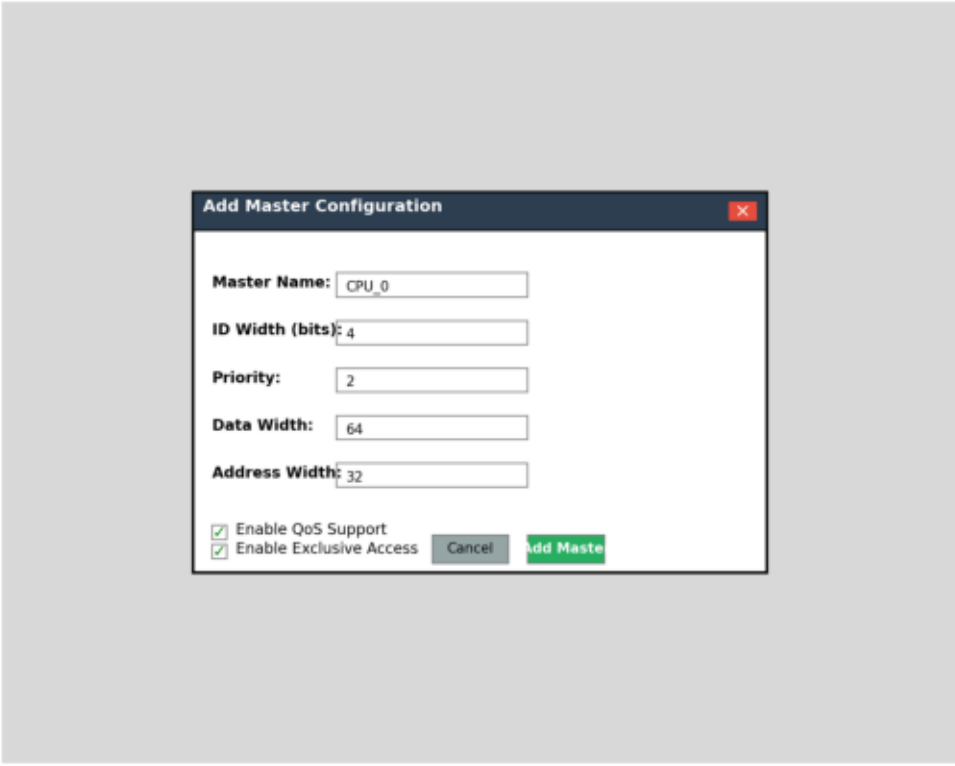
Screenshot: Step 1: Main GUI Interface Overview



This is the main AMBA Bus Matrix Configuration Tool interface. Key areas include: Left Panel (Component Library) for adding masters and slaves, Center Canvas for design layout, Right Panel (Properties) for configuration, Toolbar with essential functions like 'Add Master', 'Add Slave', 'Generate RTL', and 'Generate VIP'. The status bar shows current project state.

Step 2: Adding Bus Masters

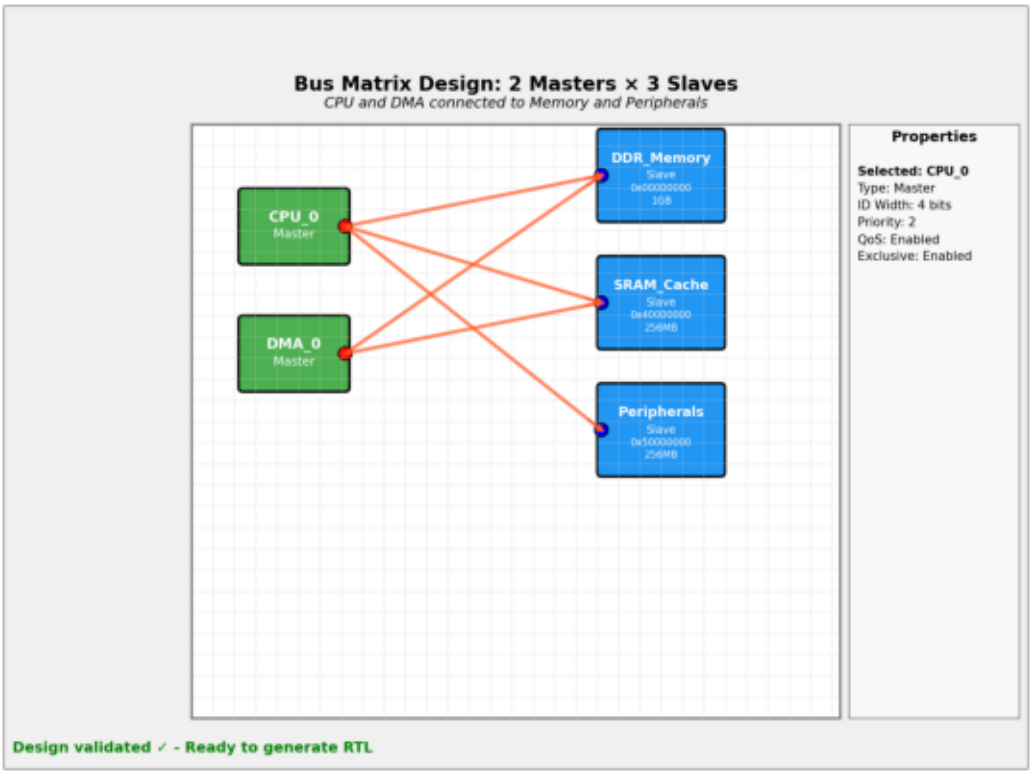
Screenshot: Step 2: Adding Bus Masters



Click 'Add Master' from the toolbar to open this configuration dialog. Set the Master Name (e.g., 'CPU_0'), configure ID Width (typically 4-8 bits), set Priority for arbitration, and enable features like QoS Support and Exclusive Access. These settings determine how the master will behave in the bus matrix. Click 'Add Master' to confirm.

Step 3: Complete Bus Matrix Design

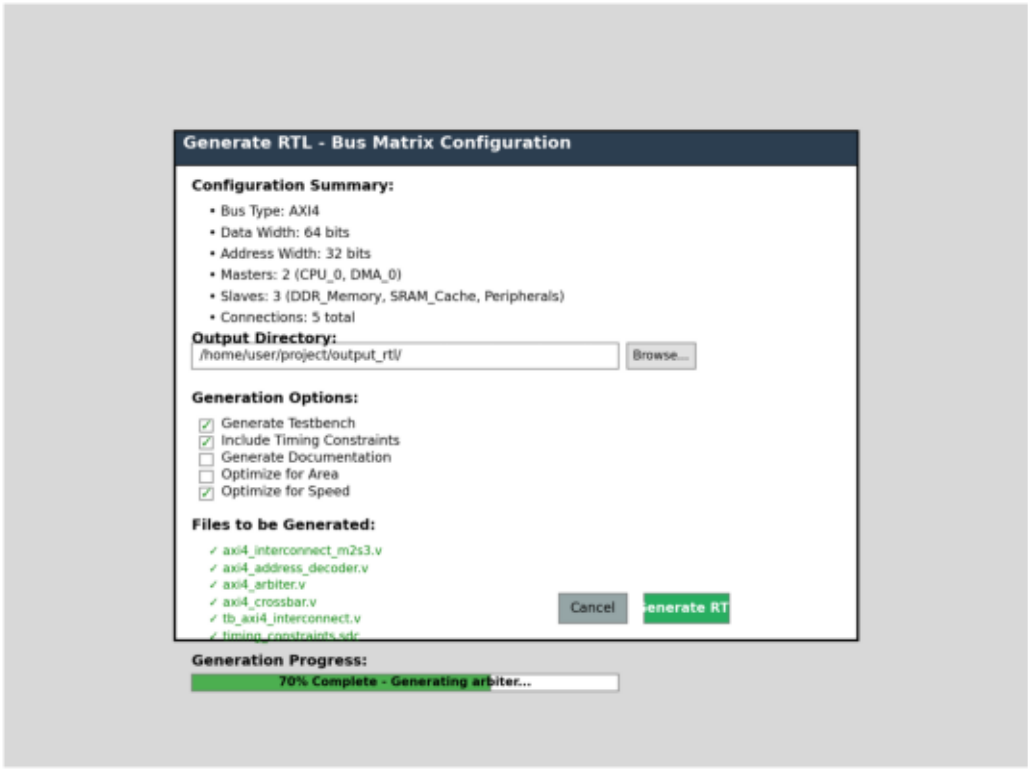
Screenshot: Step 3: Complete Bus Matrix Design



This shows a complete 2×3 bus matrix design with 2 masters (CPU_0, DMA_0) and 3 slaves (DDR_Memory, SRAM_Cache, Peripherals). Masters are shown in green, slaves in blue. Red connections indicate data paths. The Properties panel shows details for the selected component. Note the address configuration for each slave (0x00000000, 0x40000000, 0x50000000).

Step 4: RTL Generation Process

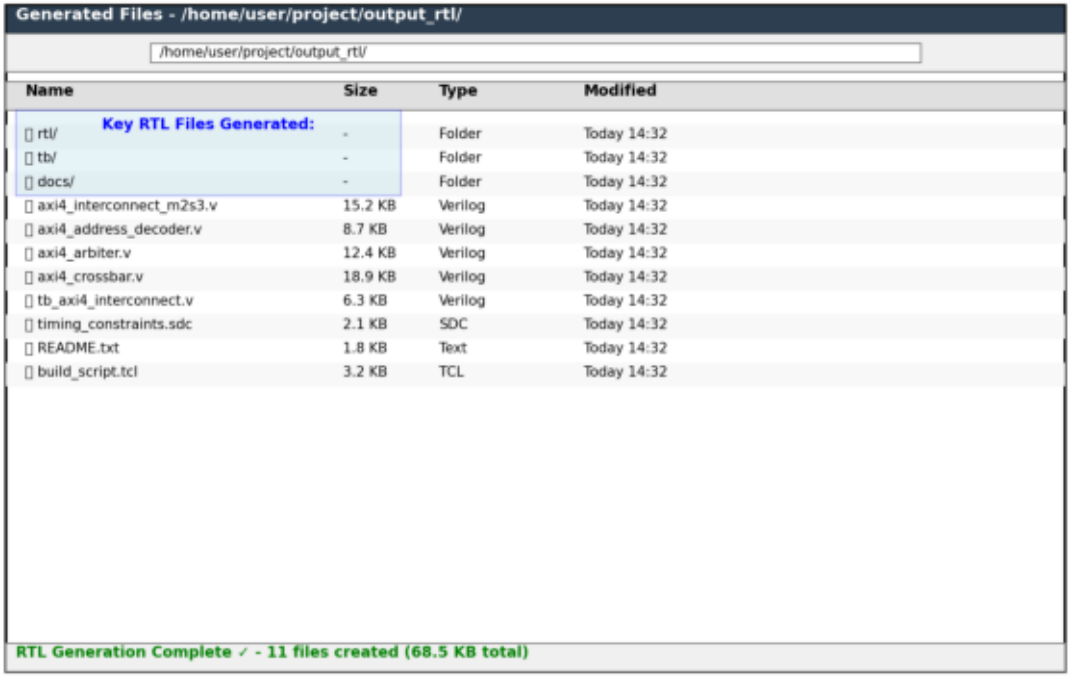
Screenshot: Step 4: RTL Generation Process



Click 'Generate RTL' to open this dialog. Review the configuration summary, set the output directory, and select generation options. The dialog shows which files will be created including the main interconnect module, address decoder, arbiter, and testbench. The progress bar indicates generation status in real-time.

Step 5: Generated RTL Files

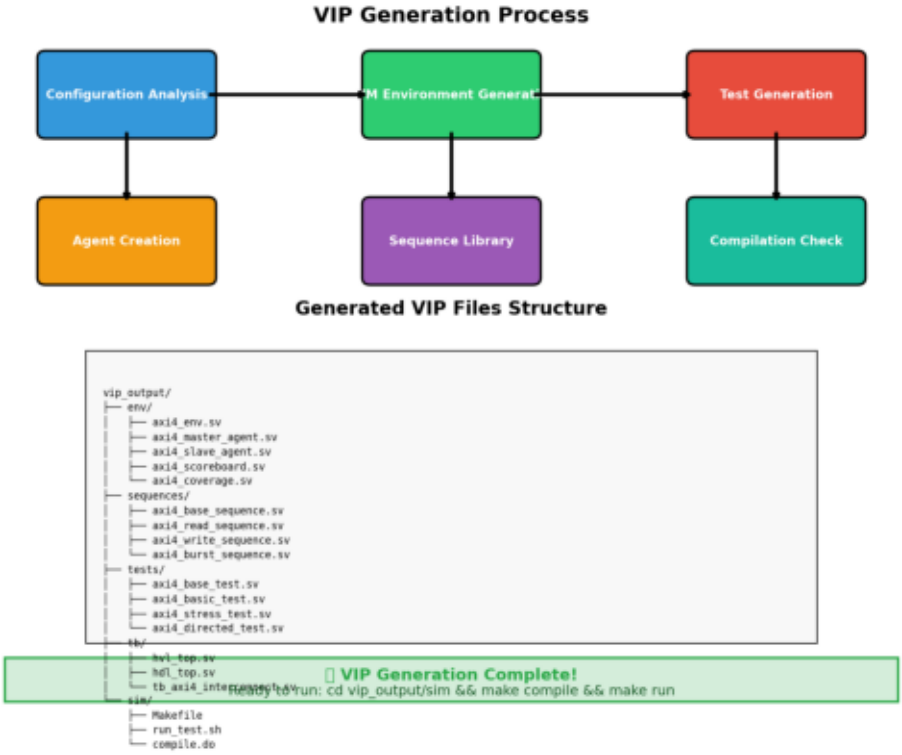
Screenshot: Step 5: Generated RTL Files



After RTL generation, this file browser shows all created files in the output directory. Key files include: axi4_interconnect_m2s3.v (main module), axi4_address_decoder.v, axi4_arbiter.v, and tb_axi4_interconnect.v (testbench). The status bar confirms successful generation. These files are ready for synthesis and simulation.

Step 6: VIP Generation Process

Screenshot: Step 6: VIP Generation Process



Click 'Generate VIP' to create a complete UVM verification environment. The process includes generating agents, sequences, tests, and testbench files. The file structure shows organized directories for env/, sequences/, tests/, tb/, and sim/. After generation, run 'cd vip_output/sim && make compile && make run' to start verification.

Complete GUI Workflow Summary

 STEP 1: Launch GUI
   • Run: ./launch_gui.sh or python3 src/bus_matrix_gui.py
   • Main window opens with canvas, toolbar, and panels

 STEP 2: Create New Project
   • File → New Project (Ctrl+N)
   • Enter project name and select AXI4 bus type
   • Set data width (typically 64 bits)

 STEP 3: Add Masters (Transaction Initiators)
   • Click "Add Master" button in toolbar
   • Configure: Name="CPU_0", ID Width=4, Priority=2
   • Enable QoS and Exclusive Access as needed
   • Repeat for additional masters (e.g., DMA_0)

 STEP 4: Add Slaves (Response Targets)
   • Click "Add Slave" button in toolbar
   • Configure addresses carefully:
     - DDR_Memory: Base=0x00000000, Size=1GB
     - SRAM_Cache: Base=0x40000000, Size=256MB
     - Peripherals: Base=0x50000000, Size=256MB

 STEP 5: Make Connections
   • Drag from master output ports to slave input ports
   • Or use Connection Matrix: View → Connection Matrix
   • Ensure all required paths are connected

 STEP 6: Validate Design
   • Tools → Validate Design (Ctrl+V)
   • Fix any address overlaps or connection issues
   • Green status indicates ready for generation

 STEP 7: Generate RTL
   • Generate → Generate RTL (Ctrl+G)
   • Choose output directory (default: output_rtl/)
   • Select options and click Generate
   • Files created: interconnect, arbiter, decoder modules

 STEP 8: Generate VIP (Optional)
   • Generate → Generate VIP (Ctrl+Shift+G)
   • Creates complete UVM verification environment
   • Output includes agents, sequences, tests, and sim scripts

 STEP 9: Run Verification
   • cd vip_output/sim
   • make compile && make run TEST=basic_test
   • View results in logs/ directory

Troubleshooting with Visual Cues

 COMMON ISSUES AND VISUAL INDICATORS:

 GUI Won't Launch
   Visual: Terminal shows "ImportError: No module named tkinter"
   Solution: sudo apt-get install python3-tk

 Address Overlap Error
   Visual: Red warning in Properties panel, status bar shows error
   Solution: Adjust slave base addresses to avoid overlap

 Connection Issues
   Visual: Disconnected ports shown with red X marks
   Solution: Drag connections from master outputs to slave inputs

 RTL Generation Fails
   Visual: Progress bar stops, error dialog appears
   Solution: Check design validation first (Tools → Validate Design)

 VIP Compilation Errors
   Visual: Error messages in simulation log files
   Solution: Set UVM_HOME environment variable, check simulator

 SUCCESS INDICATORS:

✓ Design Validated
   Visual: Green checkmark in status bar "Design validated ✓"

✓ RTL Generated
   Visual: File browser shows generated .v files with sizes

✓ VIP Ready
   Visual: Complete directory structure in vip_output/

✓ Simulation Running
   Visual: Progress messages in terminal, log files updating

 DEBUGGING TIPS:

• Enable debug mode: export AXI_VIP_DEBUG=1
• Check Properties panel for component details
• Use Connection Matrix for complex designs (View → Connection Matrix)
• Validate after each major change
• Save project frequently (Ctrl+S)

 VISUAL CHECKLIST:

□ Main window shows all panels (Library, Canvas, Properties)
□ Masters appear as green boxes with output ports
□ Slaves appear as blue boxes with input ports and addresses
□ Connections shown as red lines with arrows
□ Status bar shows "Ready" or validation status
□ Generated files appear in output directory