# Integrated Smart EDA System – Modular Architecture



## 1. Project Workspace Manager

* ProjectManager
  + CreateProject()
  + LoadProject()
  + SaveProject()
  + VersionControlManager
    - GitIntegration()
    - PermissionHandler()

## 2. Schematic Design System

* SchematicEditor
  + SymbolLibraryManager
  + DigitalSchematicCanvas
  + AnalogMixedSignalEditor
  + RTLViewer
  + HDLParser (Verilog/VHDL/SystemVerilog)
  + DRCChecker

## 3. Physical Layout Engine

* LayoutEditor
  + VectorLayoutCanvas
  + GridManager (Custom Grid, Snap-to-Atom)
  + ZoomController (Up to 1e+15%)
  + FileIOManager (GDSII, OASIS, DEF)
  + HierarchyManager
  + PCellGenerator
  + NanoStructureDesigner (CNT, Graphene, 2D Materials)

## 4. Device & Transistor Modeling

* TransistorDesigner
  + AtomicStructureEditor
  + DopingRegionDefiner
  + MaterialLibrary
  + NanoPDKBuilder

## 5. Quantum & Nano Simulation Engine

* QuantumSimulator
  + DFTSolverInterface
  + SchrödingerSolver
  + ElectronTransportAnalyzer
  + BandGapCalculator
  + SpinPhononSimulator

## 6. Circuit Simulation Engine

* SimulationManager
  + SPICEInterface
  + MixedSignalSimulator
  + AIOptimizer
  + PowerDelayAnalyzer
  + MonteCarloModule

## 7. AI-Driven Design Automation

* AIAssistant
  + AutoPlaceAndRoute
  + AIErrorPredictor
  + ReinforcementOptimizer
  + SmartDRCLVSChecker
  + CircuitGPT (Conversational Designer)

## 8. Verification & Validation

* VerificationCore
  + DRCValidator
  + LVSComparator
  + FormalVerificationEngine
  + UVMTestManager
  + FaultToleranceChecker

## 9. Testing & Yield Analysis

* TestSuiteGenerator
  + BISTLogicInserter
  + ATPGModule
  + YieldPredictor
  + StatisticalVariationAnalyzer

## 10. Fabrication Readiness Tools

* FabricationToolkit
  + MaskGenerator
  + NanoFabCompiler
  + FoundryCompatibilityChecker
  + GDSExportManager

## 11. Multi-Physics Engine

* PhysicsSimulator
  + ThermalAnalyzer
  + StressStrainSimulator
  + EMIRDropAnalyzer
  + ESDModeler
  + RFNoisePredictor

## 12. Cloud & Distributed Simulation

* CloudEngine
  + ClusterDispatcher
  + RemoteDesignViewer
  + ContainerManager (Docker/Kubernetes)
  + QuantumBackendConnector

## 13. Security & IP Protection

* SecurityManager
  + IPWatermarker
  + TrojanDetector
  + LayoutEncryptor
  + ZeroTrustModel

## 14. Visualization & Debugging Tools

* DebugTools
  + NanoViewer3D
  + TimeTraceAnalyzer
  + SignalIntegrityGrapher
  + EyeDiagramTool
  + LayoutSchematicLinker

## 15. Data Integration & Interoperability

* InteropManager
  + FormatSupport (GDS, LEF/DEF, Verilog, VHDL)
  + OpenAccessConnector
  + FoundryKitImporter
  + JSONLayoutSerializer

## 16. Documentation & Training System

* DocumentationSuite
  + AutoDocGenerator
  + InteractiveTutorials
  + AIWalkthroughAssistant
  + ReportExporter

B. Contextual Directory Stracture

smartEDA/

│

├── CMakeLists.txt # Root CMake build file

├── README.md # Project overview

├── LICENSE # License info

├── .gitignore # Git exclusions

│

├── docs/ # Documentation

│ ├── architecture.md

│ ├── api\_reference.md

│ └── tutorials/

│ └── intro\_to\_layout\_editor.md

│

├── config/ # Configurations and templates

│ ├── settings.json

│ └── themes/

│ └── darkmode.qss

│

├── external/ # External libraries and dependencies

│ ├── qt/

│ ├── eigen/

│ └── spdlog/

│

├── src/ # Main source code

│

│ ├── core/ # Core engine modules (C++)

│ │ ├── layout/ # Atomic-scale layout editor

│ │ │ ├── LayoutEngine.h/.cpp

│ │ │ ├── GridManager.h/.cpp

│ │ │ └── ZoomController.h/.cpp

│ │ ├── schematic/

│ │ │ ├── SymbolLib.h/.cpp

│ │ │ └── HDLParser.h/.cpp

│ │ ├── devices/

│ │ │ ├── TransistorModel.h/.cpp

│ │ │ └── NanoMaterial.h/.cpp

│ │ ├── simulation/

│ │ │ ├── SpiceBridge.h/.cpp

│ │ │ └── QuantumSimInterface.h/.cpp

│ │ └── verification/

│ │ ├── LVSChecker.h/.cpp

│ │ └── DRCRules.h/.cpp

│

│ ├── gui/ # GUI code (Qt)

│ │ ├── main.cpp # Application entry point

│ │ ├── MainWindow.h/.cpp

│ │ ├── widgets/

│ │ │ ├── LayoutView.h/.cpp

│ │ │ └── PropertyPanel.h/.cpp

│ │ └── qml/ # Optional if using Qt QML

│ │ └── MainView.qml

│

│ ├── ai/ # Python AI modules

│ │ ├── \_\_init\_\_.py

│ │ ├── assistant.py # AI conversational assistant

│ │ ├── optimizer.py # Auto-routing, param optimization

│ │ └── verifier.py # ML-based DRC error predictor

│

│ ├── bindings/ # C++ ↔ Python glue code

│ │ └── pybind/

│ │ └── layout\_bindings.cpp

│

│ └── utils/ # Shared utilities and helpers

│ ├── FileIO.h/.cpp

│ ├── Logger.h/.cpp

│ └── PrecisionMath.h/.cpp

│

├── test/ # Unit and integration tests

│ ├── core/

│ ├── gui/

│ ├── ai/

│ └── CMakeLists.txt

│

├── scripts/ # Development/build/deploy scripts

│ ├── setup\_env.sh

│ ├── run\_tests.py

│ └── generate\_docs.py

│

├── examples/ # Example projects and circuits

│ ├── inverter\_project/

│ └── nand\_chain/

│

└── data/ # Layout, symbol, and simulation data

├── layouts/

├── spice/

├── pdk/

└── material\_db/

To think for greater way:-

core/

└── devices/

├── NanoMaterial/

│ ├── BaseMaterial.h/.cpp # Abstract base (bandgap, density, etc.)

│ ├── GrapheneMaterial.h/.cpp # Specific graphene model

│ ├── CNTMaterial.h/.cpp # Carbon nanotube variant

│ ├── MoS2Material.h/.cpp # Molybdenum disulfide

│ ├── DopingModel.h/.cpp # Doping control and charge profiles

│ ├── QuantumDefects.h/.cpp # Point defects, vacancies, edge states

│ └── MaterialFactory.h/.cpp # Factory pattern to construct them

## Updated gui/ Directory Structure

├── gui/ # GUI frontend (Qt-based)

│

│ ├── main.cpp # Qt application entry point

│

│ ├── MainWindow.h/.cpp # Platform-independent window interface and Qt implementation

│

│ ├── widgets/ # Main reusable UI components

│ │ ├── LayoutView.h/.cpp # Layout visualization pane

│ │ ├── PropertyPanel.h/.cpp # Properties of selected components

│ │ ├── ToolboxWidget.h/.cpp # Drag-and-drop component/tool palette

│ │ ├── CanvasView.h/.cpp # Interactive view (QGraphicsView subclass)

│ │ ├── CanvasScene.h/.cpp # Drop target (QGraphicsScene subclass)

│ │ ├── CanvasItem.h/.cpp # Base class for visual items (gates, symbols)

│ │ └── ObjectPropertiesDialog.h/.cpp # Pop-up dialog to edit symbol attributes

│

│ ├── dragdrop/ # Drag-and-drop abstraction layer

│ │ ├── ComponentFactory.h/.cpp # Creates logical+visual elements from drop data

│ │ └── DragMimeTypes.h # Defines MIME type constants and helper functions

│

│ ├── qml/ # Optional modern UI via Qt Quick

│ │ └── MainView.qml # QML frontend if used

│

│ └── styles/ # Centralized styles (optional)

│ └── themes.qss # Qt stylesheet for UI consistency

| Folder / File | Purpose |

| ----------------------------------------- | --------------------------------------------------------------------- |

| `ToolboxWidget` | Left-side panel for dragging circuit components |

| `CanvasView`, `CanvasScene`, `CanvasItem` | Modular canvas system for layout/schematic work |

| `ComponentFactory` | Creates circuit objects from drops (dragged type → object logic + UI) |

| `DragMimeTypes.h` | Ensures MIME consistency for drag/drop actions |

| `ObjectPropertiesDialog` | Double-click/edit properties like name, value, width, etc. |

| `styles/themes.qss` | Optional CSS-style theme control for Qt |

Visual Concept:

+----------------------------------------------+

| Toolbox | Canvas (CanvasView) |

| ┌──────────┐ | ┌────────────────────┐ |

| │ AND Gate │──▶ │ │ |

| │ OR Gate │──▶ │ Drag here! │ |

| │ FlipFlop │──▶ │ │ |

| └──────────┘ | └────────────────────┘ |

| | [PropertyPanel] |

+----------------------------------------------+

PCB Design and Layout Tool:

├── src/

│ ├── pcbgen/ # PCB generation and design automation

│ │ ├── PCBGenerator.h/.cpp # Main pipeline

│ │ ├── NetlistToPCBConverter.h/.cpp # Map chip/schematic to PCB traces

│ │ ├── PowerPlanner.h/.cpp # Power/ground planes, decoupling, IR drop

│ │ ├── RuleEngine.h/.cpp # DRC/ERC, layer assignment

│ │ ├── PCBComponentLibrary/ # Footprints, package definitions

│ │ │ └── IC\_PMIC\_Connectors.json # e.g., footprint data for connectors, ICs, etc.

│ │ └── Exporters/

│ │ ├── GerberExporter.h/.cpp # Generates .gbr, .drl files

│ │ └── KiCadExporter.h/.cpp # Optionally export to .kicad\_pcb