# Synthesizer Documentation

The Synthesizer is the core frontend of the Tokamak zk-EVM that converts Ethereum transactions into Tokamak zk-SNARK circuits. This documentation is organized into six main sections:

### 1. [Concepts](synthesizer/synthesizer-concepts.md)

**Start here if you’re new to the Synthesizer**

Learn the fundamental concepts and high-level architecture.

### 2. [Execution Flow](synthesizer/synthesizer-execution-flow.md)

**Understand how transactions are processed**

Follow the step-by-step execution flow from transaction input to output file generation.

### 3. [Code Architecture](synthesizer/synthesizer-architecture.md)

**Deep dive into code structure and implementation**

Explore the detailed code architecture, core classes, and integration points with EthereumJS. Includes:

* Transaction processing flow
* Repository structure
* Class structure and relationships
* Practical code examples
* **Output file format reference**

### 4. [Data Structures](synthesizer/synthesizer-data-structure.md)

**Master the core data types**

Understand DataPt, StackPt, MemoryPt, and Placement - the key data structures for symbol processing.

### 5. [Opcodes](synthesizer/synthesizer-opcodes.md)

**Reference guide for EVM opcode implementation**

Detailed opcode-by-opcode reference with circuit generation details and source code links.

### 6. [Terminology](synthesizer/synthesizer-terminology.md)

**Quick reference for Synthesizer-specific terms**

Alphabetically organized glossary of key concepts, data structures, and technical terms used throughout the Synthesizer documentation.