Synopsys Learning Journeys
An easy access guide to your learning
## Table of Contents

### Product-based Journeys
- Custom Compiler
- Design Compiler NXT
- DSO.ai
- ESP
- Formality
- Fusion Compiler
- FC/ICC II Hierarchical Design Planning
- Fusion Platform Methodology
- IC Compiler II
- IC Validator
- Library Compiler
- Language
- LynxNXT
- PrimeLib
- PrimePower
- PrimeSim
- PrimeTime
- Reference Methodology
- RTL Architect
- SaberRD
- Silicon Smart
- Synplify
- StarRC
- TestMAX Advisor
- TestMAX Access
- TestMAX FuSa
- TestMAX Manager
- TestMAX SMS
- TestMAX XLBIST
- Tweaker
- ZeBu

### Role-based Journeys
- Analog Designer
- RF/Transceiver Designer
- CAD Engineer
- Digital Designer
- Physical Designer
Custom Compiler Learning Path **Analog Designer**

### Course 1: CC: Foundation I
- Custom Compiler Overview
- Library Manager
- Data I/O
- Technology
- Design Review Assistant
- Unified Constraint Management

### Course 2: CC: Schematic Entry
- Capturing Schematics
- Using Advanced Editing features
- Generating Symbols
- Schematic Entry
- Advanced Schematic Editing
- Symbol Creation
- Design Hierarchy
- Working with Text Views
- Parameterized Connections
- Schematic Overlays
- Using Estimated Parasitics

### Course 3: PrimeWave Jumpstart
- Testbench Setup
- Parametric Analysis
- Corner Analysis
- Monte Carlo Analysis
- Dynamic Device Checks

### Course 4: PrimeWave & PrimeSIM SPICE Analog Tutorial
- OP, DC, Tran Analysis
- Loop Stability Analysis
- Noise Analysis
- Transient Analysis
- Transient Noise Analysis
- S-parameter Analysis
- SOA
- MOSRA

### Course 5: CC: Reliability
- PrimeSim ResCheck Analysis
- PrimeSim EMIR Reliability Analysis
- In-Design EM Aware Layout Implementation
- In-Design Capacitance Reporting
- In-Design Resistance Checking and Reporting
- Partial Layout Extraction (needs CC Elite license)
- Layout Dependent Effects
- Voltage Dependent Rule Check

**Legend**
- Self-paced Learning
- Instructor-Led Training
- Downloadable Lab
- Cloud-based Lab
- # Duration in Days
- Badge

© 2022 Synopsys, Inc.
Custom Compiler Learning Path **RF/Transceiver Designer**

**Course 1**
- CC: Foundation I
- Custom Compiler Overview
- Library Manager
- Data I/O
- Technology
- Design Review Assistant
- Unified Constraint Management

**Course 2**
- CC: Basic Layout Design
- Layout Design Entry
- Abutment & Align Functions
- Advanced Editing Functions
- Hierarchical Design Creation
- Connectivity Engine
- Abstract Generation
- Design Rule Aware Editing
- Mcell Guard Ring
- Advanced Node Support
- Multi Pattern Technology
- User Defined Devices (UDD)
- Physical Verification
- Parasitic Extraction

**Course 3**
- CC: Accelerated Layout Design
- Schematic Driven Layout (SDL) Initialization
- SDL Placement with Connectivity Use
- SDL ECO & Cross Object Referencing
- Placement Advanced
- Symbolic Editor
- Visually Assisted Automation
- Template Manager

**Course 4**
- CC: Reliability
- PrimeSim ResCheck Analysis
- PrimeSim EMIR Reliability Analysis
- In-Design EM Aware Layout Implementation
- In-Design Capacitance Reporting
- In-Design Resistance Checking and Reporting
- Partial Layout Extraction (needs CC Elite license)
- Layout Dependent Effects
- Voltage Dependent Rule Check

**Course 5**
- CC: Automated Layout Design
- Placement Assistant
- Pin Placer
- Block Placer
- Router Introduction
- Pattern Router
- Interactive Router
- Auto Router
- Shielding
- Scripting
- Antenna Rule Support
- Technology Enablement

Legend:
- Self-paced Learning
- Instructor-Led Training
- Downloadable Lab
- Cloud-based Lab
- Duration in Days
- Badge
Custom Compiler Learning Path **CAD Engineer**

### Course 1: CC: Foundation I
- Custom Compiler Overview
- Library Manager
- Data I/O
- Technology
- Design Review Assistant
- Unified Constraint Management

### Course 2: CC: Foundation II
- Tcl API
- Customization
- Dialog Boxes
- Obtaining Database Information
- User Commands
- User Interface
- Creating Effective Tcl Scripts
- Design Migration
- Component Description Format (CDF)
- Performance Monitor
- Customizing the Netlister
- PDK Setup (for SDL)

**Legend**
- Self-paced Learning
- Instructor-Led Training
- Downloadable Lab
- Cloud-based Lab
- Duration in Days
- Badge
## Design Compiler NXT Learning Path

### Course 1
- **DC: Jumpstart**
  - Introduction
  - Design Setup
  - Reading RTL Design
  - Loading Floorplan Data
  - Constraining Design
  - Synthesis Techniques
  - Post Synthesis Outputs

### Course 2
- **DC: Synthesis NXT**
  - Introduction to DC NXT
  - Design Setup for Physical Synthesis
  - Accessing Design and Library Object
  - Constraints: Reg-to-Reg and I/O Timing
  - Advanced Schematic Editing
  - Input Transition and Output Loading
  - DC NXT Ultra Synthesis Techniques
  - Timing Analysis
  - Constraints: Multiple Clocks and Exceptions
  - SPG Flow, Congestion, Layout GUI
  - Constraints: Complex Design Considerations
  - Post-Synthesis Output Data
  - Clk Gating/Leakage
  - Power analysis

### Course 3
- **DC: Clock Gating Low Power**
  - Introduction
  - Power Analysis
  - Power Optimization
  - Clock Gating
  - Advanced Clock Gating
  - Self Gating
  - Multibit
  - ICC II Link
  - Reporting

### Course 4
- **DC: DFT Insertion**
  - Introduction
  - Scan Testing and Flows
  - Test Protocol
  - DFT Design Rule Checks
  - DFT DRC GUI Debug
  - DRC Fixing
  - Top-Down Scan Insertion
  - Advanced Scan Insertion
  - Bottom-up Scan Insertion
  - Export
  - On-Chip Clocking (OCC)
  - DFTMAX
  - Advanced Topics

### Course 5
- **DC: SystemVerilog for RTL Design**
  - Introduction
  - SystemVerilog Features
  - Implementing User Logic Intent
  - Combinational Logic/Latches
  - Meaning of full/parallel
  - Registers
  - State Machine
  - Wildcard & Tri-state logic
  - Pack/Unpack Array & Struct/Union
  - Interface & Package
  - Achieving High QoR-Coding

### Legend
- Self-paced Learning
- Instructor-Led Training
- Downloadable Lab
- Cloud-based Lab
- Duration in Days
- Badge
Recommended Learning Journey for a Digital Designer

Course 1
Library Compiler: Foundation
- Introduction
- Functional Modeling
- Timing Modeling
- Low Power Modeling
- Modeling for Test
- Library Creation Guidelines
- CCS Modeling
- OCV Modeling
- Check Library
- Advance Node Features
- Library Analytics
- Physical Library Preparation & Creation
- Fusion Library Creation

Course 2
Design Compiler NXT: Synthesis
- Introduction to DC NXT
- Design Setup for Physical Synthesis
- Accessing Design and Library Object
- Constraints: Reg-to-Reg and I/O Timing
- Advanced Schematic Editing
- Input Transition and Output Loading
- DC NXT Ultra Synthesis Techniques
- Timing Analysis
- Constraints: Multiple Clocks and Exceptions
- SPG Flow, Congestion, Layout GUI
- Constraints: Complex Design Considerations
- Post-Synthesis Output Data
- Clock Gating/Leakage Power Analysis

Course 3
Fusion Compiler: Design Creation & Synthesis
- Introduction & GUI
- Reading RTL
- Objects, Attributes, Application Options
- Compile Flows and Setup
- NDM Cell Libraries
- Loading UPF and Floorplan
- Timing Setup & OCV
- CCD Optimization
- Power Optimization
- Additional Compile Settings and Techniques

Course 4
Fusion Compiler: Design Implementation
- Floorplanning
- Setting up CTS
- Running CTS (CCD & Classic Flow)
- Routing
- Routing DRC
- Via Ladder
- Post-route Optimization
- Signoff

Course 5
Fusion Compiler: DFT Insertion
- Introduction
- Scan Testing and Flows
- Test Protocol
- DFT Design Rule Checks
- DFT DRC GUI Debug
- DRC Fixing
- Top-Down Scan Insertion
- Advanced Scan Insertion
- Bottom-up Scan Insertion
- Export
- On-Chip Clocking (OCC)
- DFTMAX
- Advanced Topics

Legend
- Self-paced Learning
- Instructor-Led Training
- Downloadable Lab
- Cloud-based Lab
- Duration in Days
- Badge
# Recommended Learning Journey for a Digital Designer

<table>
<thead>
<tr>
<th>Course 6</th>
<th>Course 7</th>
<th>Course 8</th>
<th>Course 9</th>
</tr>
</thead>
</table>

### Course 6: Initial Design Planning
- IO Planning
- From Commit to Abstract Creation
- VA and Block Shaping
- Macro Placement
- PG PPNS
- Pin Placement
- Timing and Budgeting
- Integration/Assembly

### Course 7: Prime Time: Foundation
- Introduction to STA in PrimeTime
- STA Concepts and Flow in PrimeTime
- Methodology: Qualifying Constraints
- Methodology: Generating Reports
- Constraining Multiple Clocks
- Additional Checks and Constraints
- Correlation: POCV and AWP Analysis
- Signoff: Path Based Analysis (PBA)
- Signal Integrity: Crosstalk Delay Analysis
- Signal Integrity: Crosstalk Noise Analysis
- Timing Closure: ECO/What If Analysis
- Large Data: DMSA and Hyperscale Analysis

### Course 8: StarRC: Foundation
- Extraction Fundamentals
- Gate Level Extraction
- Transistor Level Extraction
- Selective Netlist
- Field Solver
- Process Modelling
- Metal Fills
- Correlation: POCV and AWP Analysis

### Course 9: IC Validator: User (DRC & LVS)
- Setup ICV and Run DRC/LVS Testcase
- DRC Error Classification
- Execute DRC Testcase with Select Commands
- LVL
- Text Options
- Generate ICV formatted netlist
- Generate Equivalence Options
- Debug LVS Errors
- Using Short Finder

### Legend
- Self-paced Learning
- Instructor-Led Training
- Downloadable Lab
- Cloud-based Lab
- Duration in Days
- Badge
Recommended Learning Journey for a **Digital Designer**

### Course 10: IC Validator: Runset
- Overview
- Language Introduction
- Command API
- Writing a Simple "Flat" Rule Runset
- Advanced Programming Concepts
- IC Validator API Header Files
- Runset Coding Practices
- Layout Device Extraction
- Benefits of New Language
- Runset Structure
- Anatomy of Device Extraction Functions
- Property Calculation
- User-defined Property Functions

### Course 11: Reference Methodology: Jumpstart
- Introduction & Overview
- Organization & Structure
- Running RM
- Demo

### Course 12: Fusion Platform Methodology: Jumpstart
- Introduction & Overview
- Organization & Structure
- Lab Example
- Installation & Setup
- FCRM for FPM Users
- FAQ – Common Topics

---

**Legend**
- Self-paced Learning
- Instructor-Led Training
- Downloadable Lab
- Cloud-based Lab
- Duration in Days
- Badge
DSO.ai Learning Path

Course 1

DSO: Foundation

Legend

- Self-paced Learning
- Instructor-Led Training
- downloadable Lab
- Cloud-based Lab
- Duration in Days
- Badge

Introduction
Cold Start
Warm Start
ESP Learning Path

Course 1

Legend

1. Self-paced Learning
2. Instructor-Led Training
3. Downloadable Lab
4. Cloud-based Lab
5. Duration in Days
6. Badge
# Formality Learning Path

<table>
<thead>
<tr>
<th>Course 1</th>
<th>Course 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>F: Jumpstart</td>
<td>F: Foundation</td>
</tr>
<tr>
<td>Overview</td>
<td>Introduction to Equivalency checking</td>
</tr>
<tr>
<td>SVF Guidance</td>
<td>Concept &amp; Step</td>
</tr>
<tr>
<td>Design Read</td>
<td>Simple Logic Cones &amp; Failing Points</td>
</tr>
<tr>
<td>Setup for Verification</td>
<td>Multi-Stage Verification &amp; SVF</td>
</tr>
<tr>
<td>Match and Verify</td>
<td>Multi-Voltage Designs &amp; UPF</td>
</tr>
<tr>
<td>Debugging</td>
<td>Hard Verifications &amp; SVP</td>
</tr>
<tr>
<td>Formality Lab</td>
<td>Efficient Debugging</td>
</tr>
<tr>
<td></td>
<td>RTL &amp; Netlist Interpretation</td>
</tr>
<tr>
<td></td>
<td>Sequential Design Transforms &amp; SVF</td>
</tr>
<tr>
<td></td>
<td>Other Design Transforms &amp; SVF</td>
</tr>
<tr>
<td></td>
<td>Conclusion</td>
</tr>
</tbody>
</table>

## Legend

- **Self-paced Learning**
- **Instructor-Led Training**
- **Downloadable Lab**
- **Cloud-based Lab**
- **Duration in Days**
- **Badge**
# Fusion Compiler Learning Path

<table>
<thead>
<tr>
<th>Course 1</th>
<th>Course 2</th>
<th>Course 3</th>
<th>Course 4</th>
<th>Course 5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FC: Jumpstart</strong></td>
<td><strong>FC: Design Creation &amp; Synthesis</strong></td>
<td><strong>FC: Design Implementation</strong></td>
<td><strong>FC: DFT Insertion</strong></td>
<td><strong>FC: UPF</strong></td>
</tr>
<tr>
<td>Introduction</td>
<td>Introduction &amp; GUI</td>
<td>Floorplanning</td>
<td>Introduction</td>
<td>Introduction to UPF</td>
</tr>
<tr>
<td>Design Setup and Reading RTL Design</td>
<td>Reading RTL</td>
<td>Setting up CTS</td>
<td>Scan Testing and Flows</td>
<td>Power Domains</td>
</tr>
<tr>
<td>NDMs/CLIBS</td>
<td>Objects, Attributes, Application Options</td>
<td>Running CTS (CCD+classic flow)</td>
<td>Test Protocol</td>
<td>Power Strategies</td>
</tr>
<tr>
<td>Floorplan and UPF data</td>
<td>Compile Flows and Setup</td>
<td>Routing</td>
<td>DFT Design Rule Checks</td>
<td>Supply Network</td>
</tr>
<tr>
<td>Compile Flow</td>
<td>NDM Cell Libraries</td>
<td>Routing DRC</td>
<td>DFT DRC GUI Debug</td>
<td>Power States</td>
</tr>
<tr>
<td>Timing Setup and CCD</td>
<td>Loading UPF and Floorplan</td>
<td>Via Ladder</td>
<td>DRC Fixing</td>
<td>Fusion Compiler and UPF</td>
</tr>
<tr>
<td>Power Optimization</td>
<td>Timing Setup &amp; OCV</td>
<td>Post-route Optimization</td>
<td>Top-Down Scan Insertion</td>
<td></td>
</tr>
<tr>
<td>Top Level Synthesis</td>
<td>CCD Optimization</td>
<td>Signoff</td>
<td>Advanced Scan Insertion</td>
<td></td>
</tr>
<tr>
<td>Design Implementation</td>
<td>Power Optimization</td>
<td></td>
<td>Bottom-up Scan Insertion</td>
<td></td>
</tr>
<tr>
<td>Additional Compile Settings and Techniques</td>
<td>Hierarchical Synthesis</td>
<td></td>
<td>Export</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>On-Chip Clocking (OCC)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DFTMAX</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Advanced Topics</td>
<td></td>
</tr>
</tbody>
</table>

### Legend
- Self-paced Learning
- Instructor-Led Training
- Downloadable Lab
- Cloud-based Lab
- Duration in Days
- Badge

© 2022 Synopsys, Inc.
# IC Compiler II Learning Path

<table>
<thead>
<tr>
<th>Course 1</th>
<th>Course 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ICCII: Jumpstart</strong></td>
<td><strong>ICCII: Block Level Implementation</strong></td>
</tr>
<tr>
<td>Introduction</td>
<td>GUI Usage (lab)</td>
</tr>
<tr>
<td>Design/Timing Setup</td>
<td>Objects, Attributes, Application Options</td>
</tr>
<tr>
<td>NDMs/CLIBS</td>
<td>Floorplanning</td>
</tr>
<tr>
<td>Floorplan</td>
<td>Placement</td>
</tr>
<tr>
<td>Placement &amp; Optimization</td>
<td>NDM Cell Libraries</td>
</tr>
<tr>
<td>Clock Tree Synthesis</td>
<td>Design Setup</td>
</tr>
<tr>
<td>Routing</td>
<td>Timing Setup</td>
</tr>
<tr>
<td>Top Level Synthesis</td>
<td>Setting up CTS</td>
</tr>
<tr>
<td>Design Implementation</td>
<td>Running CTS (CCD+classic flow)</td>
</tr>
<tr>
<td></td>
<td>Routing</td>
</tr>
<tr>
<td></td>
<td>Routing DRC</td>
</tr>
<tr>
<td></td>
<td>Via Ladder</td>
</tr>
<tr>
<td></td>
<td>Post-route Optimization</td>
</tr>
<tr>
<td></td>
<td>Top Level Implementation</td>
</tr>
</tbody>
</table>

**Legend**
- Self-paced Learning
- Instructor-Led Training
- Downloadable Lab
- Cloud-based Lab
- Duration in Days
- Badge

---

**Design Implementation**

- Top Level Synthesis
- 1/2
- Via Ladder
- Post-route Optimization
- Top Level Implementation
# IC Validator Learning Path

## Course 1

**ICV: User (DRC & LVS)**

- Setup ICV and run DRC/LVS testcase
- DRC Error Classification
- Execute DRC testcase with select commands
- LVL
- Edtext options
- Generate ICV formatted netlist
- Generate equivalence options
- Debug LVS errors
- Using Short finder
- Using VUE & ICVWB with ICV

## Course 2

**ICV: Runset**

- Overview
- Language introduction
- Command API
- Writing a simple “flat” rule runset
- Running a simple IC Validator runset
- Advanced programming concepts
- IC Validator API header files
- Runset coding practices
- Layout device extraction
- Benefits of new language
- Runset structure
- Anatomy of device extraction functions
- Property calculation
- User-defined property functions

## Legend

- Self-paced Learning
- Instructor-Led Training
- Downloadable Lab
- Cloud-based Lab
- Duration in Days
- Badge
Library Compiler Learning Path

Course 1

LC: Foundation

- Introduction
- Functional Modeling
- Methodology: Qualifying Constraints
- Timing Modeling
- Modeling for Test
- Library Creation Guidelines
- CCS Modeling
- OCV Modeling
- check_library
- Electromigration
- 10 nm Feature
- Library Analytics
- Physical Library Preparation

Fusion Library Creation
## Language Learning Path (Page 1)

<table>
<thead>
<tr>
<th>Course 1</th>
<th>Course 2</th>
<th>Course 3</th>
<th>Course 4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>L: SVA Formal Verification</strong></td>
<td><strong>L: System Verilog Assertion</strong></td>
<td><strong>L: System Verilog for RTL Design</strong></td>
<td><strong>L: System Verilog Testbench</strong></td>
</tr>
<tr>
<td>Introduction to SVA</td>
<td>Introduction</td>
<td>Basic System Verilog Features</td>
<td>The Device Under Test</td>
</tr>
<tr>
<td>Formal Testbench</td>
<td>Types of Assertions</td>
<td>Implementing User Logic Intent (combinatorial logic &amp; latch)</td>
<td>System Verilog Verification Environment</td>
</tr>
<tr>
<td>Coding Recommendation – Do's and Don'ts</td>
<td>Action Blocks</td>
<td>Implementing User Logic Intent (meaning of full and parallel)</td>
<td>System Verilog Testbench Language Basics - 1</td>
</tr>
<tr>
<td>Resources</td>
<td>Disabling/Combining/Embedding Assertions</td>
<td>Implementing User Logic Intent (implementing registers)</td>
<td>System Verilog Testbench Language Basics - 2</td>
</tr>
<tr>
<td></td>
<td>Controlling Assertions</td>
<td>Implementing User Logic Intent (implementing state machines)</td>
<td>Managing Concurrency in System Verilog</td>
</tr>
<tr>
<td></td>
<td>Sequences and Sequence Repetition</td>
<td>Implementing User Logic Intent (wildcard and tri-state logic)</td>
<td>Object-Oriented Programming: Encapsulation</td>
</tr>
<tr>
<td></td>
<td>Sequences Operators</td>
<td>Advanced System Verilog Features (packed or unpacked array and struct)</td>
<td>Object-Oriented Programming: Randomization</td>
</tr>
<tr>
<td></td>
<td>Synthesis Assertion Coverage</td>
<td>Advanced System Verilog Features (System Verilog interface)</td>
<td>Voltage Dependent Rule Check</td>
</tr>
<tr>
<td></td>
<td>Assertion Libraries</td>
<td>Advanced System Verilog Features (System Verilog package)</td>
<td>Object-Oriented Programming: Inheritance</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Inter-Thread Communications</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Functional Coverage</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>System Verilog UVM preview</td>
</tr>
</tbody>
</table>

### Resources

- **Coding Recommendation – Do's and Don'ts**
- **Introduction to SVA**
- **Formal Testbench**
- **Self-paced Learning**
- **Instructor-Led Training**
- **Downloadable Lab**
- **Cloud-based Lab**
- **Duration in Days**
- **Badge**
## Language Learning Path (Page 2)

### Course 5

**L: System Verilog Verification using UVM**

<table>
<thead>
<tr>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Verilog OOP</td>
</tr>
<tr>
<td>Inheritance Review</td>
</tr>
<tr>
<td>UVM Structural Overview</td>
</tr>
<tr>
<td>Modeling Stimulus (UVM Transactions)</td>
</tr>
<tr>
<td>Creating Stimulus Sequences (UVM Sequence)</td>
</tr>
<tr>
<td>Component Configuration and Factory</td>
</tr>
<tr>
<td>TLM Communications</td>
</tr>
<tr>
<td>Scoreboard &amp; Coverage</td>
</tr>
<tr>
<td>UVM Callback</td>
</tr>
<tr>
<td>Advance Sequence/Sequencer</td>
</tr>
<tr>
<td>Phasing and Objections</td>
</tr>
<tr>
<td>Register Layer Abstraction (RAL)</td>
</tr>
<tr>
<td>Summary</td>
</tr>
</tbody>
</table>

### Legend

- Self-paced Learning
- Instructor-Led Training
- Downloadable Lab
- Cloud-based Lab
- Duration in Days
- Badge
LynxNXT Learning Path

Course 1

Legend

- Self-paced Learning
- Instructor-Led Training
- Downloadable Lab
- Cloud-based Lab
- Duration in Days
- Badge

LNXT: Foundation

- Introduction
- Variable Editor
- Flow Editor
- Execution Monitor
- Failure Debug
- Exploration
- Command Line Interface
- Working with FPM
PrimeLib Learning Path

Course 1

PL: Foundation

Tool Introduction
Global Setting to start characterization
Cell Level Setting to Configure Arcs
Different Characterization flow
Creating multiple Connect Database
Debugging and Troubleshooting
Complex Cell Characterization
Timing Characterization
Constraint Timing Characterization
Power Characterization

Legend

Self-paced Learning
Instructor-Led Training
Downloadable Lab
Cloud-based Lab
Duration in Days
Badge
## PrimePower Learning Path

### Course 1: PP: Jumpstart
- Power Analysis Input
- Power Components
- Leakage Power
- Internal Power
- Switching Power
- Inputs & Outputs of Power Analysis
- Simulation Activity Files
- Flow & Report

### Course 2: PP: Foundation
- Introduction
- Power Analysis
- Power Components
- Leakage Power
- Internal Power
- Switching Power
- Leakage & Internal Power Data
- Input & Outputs of Power Analysis
- PrimePower Analysis Modes
- Simulation Activity Files
- RTL Activity Flow
- Gate-Level Activity Flow
- PrimePower Analysis Accuracy
- PrimePower Standalone – ASCII Flow

### PrimePower -- Session Based Flow
- Check & Report Power
- Report Switching Activity
- Summary

### Legend
- Self-paced Learning
- Instructor-Led Training
- Downloadable Lab
- Cloud-based Lab
- Duration in Days
- Badge
# PrimeSim Learning Path

<table>
<thead>
<tr>
<th>Course 1</th>
<th>Course 2</th>
<th>Course 3</th>
<th>Course 4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PS: Jumpstart</strong></td>
<td><strong>PS: Foundation</strong></td>
<td><strong>PS: Advance</strong></td>
<td><strong>PS: Advance</strong></td>
</tr>
<tr>
<td>PrimeSim Static CCK Introduction</td>
<td>PrimeSim XA Introduction</td>
<td>CCK Advanced ERC and ESD Checks</td>
<td>Interactive Mode</td>
</tr>
<tr>
<td>PrimeSim CCK Built-In Checks</td>
<td>PrimeSim XA Nelist format Support</td>
<td>CCK Propagation Engine – XPL and Analog Propagation</td>
<td>Distributed Processing</td>
</tr>
<tr>
<td>PrimeSim CCK Interactive Debugging Commands</td>
<td>PrimeSim XA Analyses support</td>
<td>CCK Custom Programmable Checks</td>
<td>Monte-Carlo (MC)</td>
</tr>
<tr>
<td>PrimeSim CCK False Error Pruning</td>
<td>Command/Option Usage &amp; Precedence Rules</td>
<td>Custom Check Assertion</td>
<td>MOSRA</td>
</tr>
<tr>
<td>Custom Compiler – PrimeSim CCK : Setup and run</td>
<td>PrimeSim XA Post-Layout Simulation</td>
<td>GUI : Cross-Probe, Filtering, Waiver, Grouping, export</td>
<td>Aging and Self-Heating</td>
</tr>
<tr>
<td>PrimeSim XA Command Line Usage</td>
<td>PrimeSim XA Log File Details</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Accuracy and Speed Trade-off</td>
<td>Back-Annotation &amp; XBA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Probing in PrimeSim XA</td>
<td>PrimeSim XA .ALTER Usage</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PrimeSim XA .DATA Usage</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Legend
- **Self-paced Learning**
- **Instructor-Led Training**
- **Downloadable Lab**
- **Cloud-based Lab**
- **Duration in Days**
- **Badge**
# PrimeTime Learning Path

## Course 1
**PT: Jumpstart**
- Overview
- PrimeTime Implementation Flow
- PrimeTime Inputs & Outputs
- Timing Analysis Flow
- Load Design & Check
- Load Library & Check
- Read Parasitic & Check
- Source Constraints & Check
- Constraints Completeness
- Coverage Analysis
- Report
- Saving & Exit

## Course 2
**PT: Foundation**
- Introduction to STA in PrimeTime
- STA Concepts and Flow in PrimeTime
- Methodology: Qualifying Constraints
- Methodology: Generating Reports
- Constraining Multiple Clocks
- Additional Checks and Constraints
- Correlation: POCV and AWP Analysis
- Signoff: Path Based Analysis (PBA)
- Signal Integrity: Crosstalk Delay Analysis
- Signal Integrity: Crosstalk Noise Analysis
- Timing Closure: ECO/What If Analysis
- Large Data: DMSA and Hyperscale Analysis

## Course 3
**PT: HyperScale**
- Introduction HyperScale
- Flat Context Flow
- Bottom Up Flow
- Generating HyperScale Block Models
- Constraint Consistency
- Clock Mapping
- HyperScale Top-Down Flow
- HyperScale-Driven ECO
- Summary

## Course 4
**PT: Scalable STA**
- Hierarchical Methodologies
- HyperScale
- HyperScale Hybrid Flow
- Distributed & Scenario Analysis
- HyperGrid
- DMSA
- DVFA/SMVA
- PBA Technologies
- Best Practices

---

**Legend**
- Self-paced Learning
- Instructor-Led Training
- Downloadable Lab
- Cloud-based Lab
- Duration in Days
- Badge
Recommend Learning Journey: **Physical Designer**

**Course 1**
- Introduction
- Design Setup and Reading RTL Design
- NDMs/CLIBS
- Floorplan and UPF data
- Compile Flow
- Timing Setup and CCD
- Power Optimization
- Top Level Synthesis
- Design Implementation

**Course 2**
- Introduction & GUI
- Reading RTL
- Objects, Attributes, Application Options
- Compile Flows and Setup
- NDM Cell Libraries
- Loading UPF and Floorplan
- Timing Setup & OCV
- CCD Optimization
- Power Optimization
- Additional Compile Settings and Techniques
- Hierarchical Synthesis

**Course 3**
- Floorplanning
- Setting up CTS
- Running CTS (CCD+classic flow)
- Routing
- Routing DRC
- Via Ladder
- Post-route Optimization
- Signoff

**Course 4**
- Introduction
- Scan Testing and Flows
- Test Protocol
- DFT Design Rule Checks
- DFT DRC GUI Debug
- DRC Fixing
- Top-Down Scan Insertion
- Advanced Scan Insertion
- Bottom-up Scan Insertion
- Export
- On-Chip Clocking (OCC)
- DFTMAX
- Advanced Topics

**Course 5**
- Initial Design Planning
- IO Planning
- From Commit to Abstract Creation
- VA and Block Shaping
- Macro Placement
- PG PPNS
- Pin Placement
- Timing and Budgeting
- Integration/Assembly

---

**Legend**
- Self-paced Learning
- Instructor-Led Training
- Downloadable Lab
- Cloud-based Lab
- Duration in Days
- Badge
Recommend Learning Journey: **Physical Designer**

### Course 6
- FC: MV/UPF
- Introduction UPF
- Power Domains
- Power Strategies
- Supply Network
- Power States
- Fusion Compiler and UPF

### Course 7
- RM: Jumpstart
- Introduction & Overview
- Organization & Structure
- Running RM
- Demo

---

**Legend**
- Self-paced Learning
- Instructor-Led Training
- Downloadable Lab
- Cloud-based Lab
- Duration in Days
- Badge
Reference Methodology Learning Path

Course 1

RM: Jumpstart

Introduction & Overview
Organization & Structure
Running RM
Demo

Legend

Self-paced Learning
Instructor-Led Training
Downloadable Lab
Cloud-based Lab
Duration in Days
Badge
# RTL Architect Learning Path

## Course 1
- **RTLA: Jumpstart**
  - Introduction & Overview
  - RTL Architect Key Features
  - Predictive Engine
  - Unified GUI
  - Physical Floorplanning
  - Power Analysis
  - Logic Restructuring
  - Constraint Management
  - Flows
  - rtl_opt Mega Command
  - Block-Level Flow/Breakpoints
  - Hierarchical Flow/Breakpoints

## Course 2
- **RTLA: Using RTL Restructuring**
  - RTL Restructuring
  - Group
  - Ungroup
  - Reparent
  - Restructured RTL, SDC, UPF, SAIF Generation
  - Reparenting and Writing RTL
  - Demo

---

### Legend
- Self-paced Learning
- Instructor-Led Training
- Downloadable Lab
- Cloud-based Lab
- Duration in Days
- Badge

---

*Synopsys Confidential Information*
StarRC Learning Path

Course 1
- SRC: Jumpstart
- Interconnect
- Coupling Capacitance
- Classes of Extractors
- Input for Parasitic Extraction
- StarRC Flow
- SMC Flow
- ITF File

Course 2
- SRC: Foundation
- Extraction Fundamentals
- Gate level Extraction
- Transistor Level Extraction
- Selective Netlist
- Field Solver
- Process Modelling
- Metal Fills

Legend
- Self-paced Learning
- Instructor-Led Training
- Downloadable Lab
- Cloud-based Lab
- Duration in Days
- Badge

Synopsys Confidential Information
# SaberRD Learning Path

## Course 1

**SRC: Foundation Training Series**

<table>
<thead>
<tr>
<th>Module</th>
<th>Duration in Days</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timing Domain Analysis</td>
<td>3</td>
</tr>
<tr>
<td>Schematic Capture &amp; Parts Library</td>
<td></td>
</tr>
<tr>
<td>Operating Points &amp; Small Signal Frequency Analysis</td>
<td></td>
</tr>
<tr>
<td>Test Automation</td>
<td></td>
</tr>
<tr>
<td>Design Optimization</td>
<td></td>
</tr>
<tr>
<td>Introduction to Modeling</td>
<td></td>
</tr>
<tr>
<td>Import SPICE Models</td>
<td></td>
</tr>
<tr>
<td>Modeling with Table Look-Up</td>
<td></td>
</tr>
<tr>
<td>Modeling with StateAMS</td>
<td></td>
</tr>
<tr>
<td>Robust Design &amp; Sensitivity Analysis</td>
<td></td>
</tr>
<tr>
<td>Monte Carlo &amp; Pareto</td>
<td></td>
</tr>
<tr>
<td>Worst-Case Analysis</td>
<td></td>
</tr>
<tr>
<td>Fault Analysis</td>
<td></td>
</tr>
<tr>
<td>Stress Analysis</td>
<td></td>
</tr>
</tbody>
</table>

**Legend**

- Self-paced Learning
- Instructor-Led Training
- Downloadable Lab
- Cloud-based Lab
- Badge

© 2022 Synopsys, Inc.
Silicon Smart: Learning Path

Course 1

SS: Foundation

<table>
<thead>
<tr>
<th>Introduction and Overview</th>
<th>User Defined Stimulus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Re-Characterization and Full New Characterization Flows</td>
<td>Model API</td>
</tr>
<tr>
<td>Char-point Directory Structure and Important Configuration files</td>
<td>AOCV Modeling</td>
</tr>
<tr>
<td>HDL Analyst</td>
<td>IBIS Modeling</td>
</tr>
<tr>
<td>Data Caching Mechanism</td>
<td></td>
</tr>
<tr>
<td>Arc Selection and State Dependent Measurements</td>
<td></td>
</tr>
<tr>
<td>Methods for defining Complex Cells</td>
<td></td>
</tr>
<tr>
<td>Characterization Methodology for Level Shifters</td>
<td></td>
</tr>
<tr>
<td>Characterization Methodologies for various SiliconSmart measurements</td>
<td></td>
</tr>
<tr>
<td>Debugging with VCS</td>
<td></td>
</tr>
<tr>
<td>Debugging and Troubleshooting Failures</td>
<td></td>
</tr>
<tr>
<td>Validation Utilities</td>
<td></td>
</tr>
</tbody>
</table>

Legend

Self-paced Learning
Instructor-Led Training
Downloadable Lab
Cloud-based Lab
Duration in Days
Badge

© 2022 Synopsys, Inc.
Synplify: Learning Path

Course 1

S: Foundation

- Introduction to Synplify Elite Flow
- Creating and Running Synplify Project
- View Log File
- HDL Analyst
- Handling High Reliability Designs
- Implementing Fault Tolerant FSMs
- ECC RAM Inferring
- Importing Quartus IP in Synplify Projects
- Debugging with SpyGlass
- Debugging with VCS
- Identify Instrumentor and Debugger

Legend

- Self-paced Learning
- Instructor-Led Training
- Downloadable Lab
- Cloud-based Lab
- Duration in Days
- Badge
TestMAX Advisor Learning Path

Course 1

TMA: Jumpstart

Early Testability Goals and Reports
Debug using the GUI
Transition Delay Checks
Random Resistant Fault Analysis and Test Points
Post stitch DRC Checks
Connectivity Checks
Flow With TestMAX Manager

Legend

Self-paced Learning
Instructor-Led Training
Downloadable Lab
Cloud-based Lab
Duration in Days
Badge
# TestMAX Access Learning Path

**Course 1**
- TMA: Jumpstart
  - TestMAX Access structure
  - IEEE 1687 interface to drive internal instruments through TDR
  - SIBs
  - Define Ring configuration
  - Definition of Server and Subserver
  - AIT
  - PDL Pattern Porting
  - PDL data packetization
  - Validation of AIT

**Course 2**
- TMA: ATPG & Diagnosis
  - Manufacturing Test and ATPG
  - Building ATPG Models
  - Running DRC
  - Fault Models and Managing Faults
  - Controlling ATPG
  - Post ATPG Analysis
  - Pattern Validation
  - At-Speed Testing and Constraints
  - Transition Delay Testing
  - On-Chip Clocking and Compression
  - Diagnosis

---

**Legend**
- Self-paced Learning
- Instructor-Led Training
- Downloadable Lab
- Cloud-based Lab
- Duration in Days
- Badge
TestMAX FuSa Learning Path

Course 1

TMF: Jumpstart

Legend:
- Self-paced Learning
- Instructor-Led Training
- Downloadable Lab
- Cloud-based Lab
- Duration in Days
- Badge

Functional Safety for Automotive Designs
TestMAX FuSa: Introduction
TestMAX FuSa: Static FuSa Analysis
Running TestMAX FuSa: Requirement & Constraints
Functional Safety reporting in TestMAX FuSa
TestMAX Manager Learning Path

Course 1

TMM: Jumpstart

- Launching and Configuring tool
- Objects, Attributes, Application Options
- NDM Cell Libraries
- Timing Setup

Legend

- Self-paced Learning
- Instructor-Led Training
- Downloadable Lab
- Cloud-based Lab
- Duration in Days
- Badge
TestMAX SMS Learning Path

Course 1

**TM-SMS: Architecture**

- Introduction
- SMS Wrapper
- SMS Processor
- MMB Processor
- SMS Server
- Conclusion

**Legend**

- Self-paced Learning
- Instructor-Led Training
- downloadable Lab
- Cloud-based Lab
- Duration in Days
- Badge
## TestMAX XLBIST Learning Path

### Course 1

**TMXB: Jumpstart**

<table>
<thead>
<tr>
<th>LogicBIST basics</th>
<th>Troubleshooting and Debug hints</th>
</tr>
</thead>
<tbody>
<tr>
<td>XLBIST architecture</td>
<td>Intro to AIT</td>
</tr>
<tr>
<td>XLBIST and SEQ modes of operation</td>
<td></td>
</tr>
<tr>
<td>IEEE 1500 I/F and internal resources</td>
<td></td>
</tr>
<tr>
<td>XLBIST patterns and interval definition</td>
<td></td>
</tr>
<tr>
<td>Random Resistant Fault analysis and Test Point insertion</td>
<td></td>
</tr>
<tr>
<td>X propagation analysis and fixing</td>
<td></td>
</tr>
<tr>
<td>OCC and Clock Weights</td>
<td></td>
</tr>
<tr>
<td>Reset Controller and Reset Weights</td>
<td></td>
</tr>
<tr>
<td>Programmable SE Timing Margin</td>
<td></td>
</tr>
<tr>
<td>Remap XLBIST Patterns for Debug and Diagnosis</td>
<td></td>
</tr>
<tr>
<td>Validation of XLBIST patterns</td>
<td></td>
</tr>
<tr>
<td>Porting of XLBIST patterns</td>
<td></td>
</tr>
<tr>
<td>Simulation steps for validation</td>
<td></td>
</tr>
</tbody>
</table>

**Legend**

- Self-paced Learning
- Instructor-Led Training
- Downloadable Lab
- Cloud-based Lab
- Duration in Days
- Badge
Tweaker Learning Path

Course 1

TW: ECO

ECO Flow and Interoperability
Basic ECO Flow
Timing ECO
Useful Skew Clock ECO
Power ECO
Area Recovery
Reliability Recovery
Advanced ECO Features
Hierarchy Design Flow

Legend

- Self-paced Learning
- Instructor-Led Training
- Downloadable Lab
- Cloud-based Lab
- Duration in Days
- Badge
Zebu Learning Path

Course 1

- Zebu: Foundation
  - Introduction to Emulation
  - ZeBu overview (HW and SW)
  - ZeBu Ecosystem
  - ZeBu Compile
  - ZeBu Runtime
  - Tuning ZeBu for High Performance / TAT / Capacity
  - Gate Level emulation

Course 2

- Zebu: Advance
  - Transactors – Guide to integration + List
  - Low Power emulation
  - Virtual Host and Devices
  - ZeBu Debug
  - Hybrid Emulation
  - ZeBu Empower – SW based power analysis
  - Real world interfaces with speed adaptors

Legend

- Self-paced Learning
- Instructor-Led Training
- Downloadable Lab
- Cloud-based Lab
- Duration in Days
- Badge