Synopsys Learning Journeys
An easy access guide to your learning
CONFIDENTIAL INFORMATION

The information contained in this presentation is the confidential and proprietary information of Synopsys. You are not permitted to disseminate or use any of the information provided to you in this presentation outside of Synopsys without prior written authorization.

IMPORTANT NOTICE

In the event information in this presentation reflects Synopsys’ future plans, such plans are as of the date of this presentation and are subject to change. Synopsys is not obligated to update this presentation or develop the products with the features and functionality discussed in this presentation. Additionally, Synopsys’ services and products may only be offered and purchased pursuant to an authorized quote and purchase order or a mutually agreed upon written contract with Synopsys.
# 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
- HAPS
- IC Compiler II
- IC Validator
- Library Compiler
- Language
- LynxNXT
- PrimeLib
- PrimePower
- PrimeSim
- PrimeTime
- Reference Methodology
- RTL Architect
- SaberRD
- Synplify
- StarRC
- TestMAX Advisor
- TestMAX Access
- TestMAX FuSa
- TestMAX Manager
- TestMAX SMS
- TestMAX XLBIST
- Tweaker
- VC Formal
- ZeBu

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

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

**Course 2**: Custom Compiler: 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 Parasitic

**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
- Transient + Monte Carlo Analysis
- MOSRA

**Course 5**: Custom Compiler: 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

© 2023 Synopsys, Inc.
Custom Compiler Learning Path **RF/Transceiver Designer**

Course 6

- Custom Compiler : ICC2 CoDesign
- Data Preparation
- Analog BlackBox Preparation
- Digital Implementation
- ICC2 Editing
- Capacitance Reporting and Checking

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
- Clock Gating/Leakage Power Analysis

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

Synopsys Confidential Information
### 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>
<tbody>
<tr>
<td><strong>Fusion Compiler: SOC Design Planning</strong></td>
<td><strong>Prime Time: Foundation</strong></td>
<td><strong>StarRC: Foundation</strong></td>
<td><strong>IC Validator: User (DRC &amp; LVS)</strong></td>
</tr>
<tr>
<td>Initial Design Planning</td>
<td>Introduction to STA in PrimeTime</td>
<td>Extraction Fundamentals</td>
<td>Setup ICV and Run DRC/LVS Testcase</td>
</tr>
<tr>
<td>IO Planning</td>
<td>STA Concepts and Flow in PrimeTime</td>
<td>Gate Level Extraction</td>
<td>DRC Error Classification</td>
</tr>
<tr>
<td>From Commit to Abstract Creation</td>
<td>Methodology: Qualifying Constraints</td>
<td>Transistor Level Extraction</td>
<td>Execute DRC Testcase with Select Commands</td>
</tr>
<tr>
<td>VA and Block Shaping</td>
<td>Methodology: Generating Reports</td>
<td>Selective Netlist</td>
<td>LVL</td>
</tr>
<tr>
<td>Macro Placement</td>
<td>Constraining Multiple Clocks</td>
<td>Field Solver</td>
<td>Text Options</td>
</tr>
<tr>
<td>PG PPNS</td>
<td>Additional Checks and Constraints</td>
<td>Process Modelling</td>
<td>Generate ICV formatted netlist</td>
</tr>
<tr>
<td>Pin Placement</td>
<td>Correlation: POCV and AWP Analysis</td>
<td>Metal Fills</td>
<td>Generate Equivalence Options</td>
</tr>
<tr>
<td>Timing and Budgeting</td>
<td>Signoff: Path Based Analysis (PBA)</td>
<td></td>
<td>Debug LVS Errors</td>
</tr>
<tr>
<td>Integration/Assembly</td>
<td>Signal Integrity: Crosstalk Delay Analysis</td>
<td></td>
<td>Using Short Finder</td>
</tr>
<tr>
<td></td>
<td>Signal Integrity: Crosstalk Noise Analysis</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Timing Closure: ECO/What If Analysis</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Large Data: DMSA and Hyperscale Analysis</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**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
- 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

## 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

- Introduction
- Cold Start
- Warm Start

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

Course 1

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

ESP: Jumpstart

- Introduction
- Symbolic Simulations
- Functional Accuracy
- PIV Introduction
# Formality Learning Path

## Course 1
- **Formality: Jumpstart**
  - Overview
  - SVF Guidance
  - Design Read
  - Setup for Verification
  - Match and Verify
  - Debugging
  - Formality Lab

## Course 2
- **Formality: Foundation**
  - Introduction to Equivalency checking
  - Concept & Step
  - Simple Logic Cones & Failing Points
  - Multi-Stage Verification & SVF
  - Multi-Voltage Designs & UPF
  - Hard Verifications & SVP
  - Efficient Debugging
  - RTL & Netlist Interpretation
  - Sequential Design Transforms & SVF
  - Other Design Transforms & SVF
  - Conclusion

## 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>Fusion Compiler: Jumpstart</strong></td>
<td><strong>Fusion Compiler: Design Creation &amp; Synthesis</strong></td>
<td><strong>Fusion Compiler: Design Implementation</strong></td>
<td><strong>Fusion Compiler: DFT Synthesis</strong></td>
<td><strong>Fusion Compiler: 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>Fusion Compiler Reporting</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></td>
<td>Additional Compile Settings and Techniques</td>
<td></td>
<td>Export</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Hierarchical Synthesis</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

© 2023 Synopsys, Inc.
FC/ICC II Hierarchical Design Planning Learning Path

Course 1

Fusion Compiler: SOC
Design Planning

- 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
Fusion Platform Methodology Learning Path

Course 1

Legend

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

Introduction & Overview
Organization & Structure
Lab Example
Installation & Setup
FCRM for FPM Users
FAQ – Common Topics
HAPS® Hardware Learning Path

### Course 1
- HAPS-80: Hardware and ProtoCompiler
  - Hardware Overview
  - System Clocks
  - HapsTrak3 Connectors
  - Daughter Board and Cables
  - Rack Mounting Solution
  - Confpro
  - HAPS ProtoCompiler Flow
  - Database Concepts
  - HAPS ProtoCompiler Flow Overview
  - Graphical User Interface
  - Introduction to Debug

### Course 2
- HAPS-100: Hardware and ProtoCompiler
  - Hardware Overview
  - System Clocks
  - HapsTrak3 Connectors
  - Daughter Boards and Cables
  - UMRBus 3.0 Overview
  - HAPS ProtoCompiler Flow
  - Database Concepts
  - HAPS ProtoCompiler Flow Overview
  - Graphical User Interface
  - Introduction to Debug

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

Course 1

IC Compiler II: Jumpstart

- Introduction
- Design/Timing Setup
- NDMs/CLIBS
- Floorplan
- Placement & Optimization
- Clock Tree Synthesis
- Routing
- Top Level Synthesis
- Design Implementation

Course 2

IC Compiler II: Block Level Implementation

- GUI Usage (lab)
- Objects, Attributes, Application Options
- Floorplanning
- Placement
- NDM Cell Libraries
- Design Setup
- Timing Setup
- Setting up CTS
- Running CTS (CCD+classic flow)
- Routing
- Routing DRC
- Via Ladder
- Post-route Optimization
- Top Level Implementation
- Signoff

Legend

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

**Course 1**
- **IC Validator: User** (DRC & LVS)
  - Setup IC and run DRC/LVS testcase
  - DRC Error Classification
  - Execute DRC testcase with select commands
  - LVL
  - Editext options
  - Generate ICV formatted netlist
  - Generate equivalence options
  - Debug LVS errors
  - Using Short finder
  - Using VUE & ICVWB with ICV

**Course 2**
- **IC Validator: 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

**Course 3**
- **IC Validator: DRC Runset**
  - Compare
  - Benefits of new language
  - PXL compare syntax strategy
  - Anatomy of compare functions
  - Complementary compare functions
  - User-defined functions
  - StarRC transistor-level extraction flow
  - Writing a basic "single file" runset
  - Writing debug output to OASIS/GDS Using Layer Debugger
  - Understanding Error Messages & Using Diagnostic Functions
  - Options Functions

---

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

© 2023 Synopsys, Inc.
Library Compiler Learning Path

Course 1

Library Compiler: Foundation

<table>
<thead>
<tr>
<th>Introduction</th>
<th>Fusion Library Creation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional Modeling</td>
<td></td>
</tr>
<tr>
<td>Methodology: Qualifying Constraints</td>
<td></td>
</tr>
<tr>
<td>Timing Modeling</td>
<td></td>
</tr>
<tr>
<td>Modeling for Test</td>
<td></td>
</tr>
<tr>
<td>Library Creation Guidelines</td>
<td></td>
</tr>
<tr>
<td>CCS Modeling</td>
<td></td>
</tr>
<tr>
<td>OCV Modeling</td>
<td></td>
</tr>
<tr>
<td>check_library</td>
<td></td>
</tr>
<tr>
<td>Electromigration</td>
<td></td>
</tr>
<tr>
<td>10 nm Feature</td>
<td></td>
</tr>
<tr>
<td>Library Analytics</td>
<td></td>
</tr>
<tr>
<td>Physical Library Preparation</td>
<td></td>
</tr>
</tbody>
</table>
Language Learning Path

Course 1
Language: SVA Formal Verification
- Introduction to SVA
- Formal Testbench
- Coding Recommendation – Do’s and Don’ts
- Resources

Course 2
Language: System Verilog Assertion
- Introduction
- Types of Assertions
- Action Blocks
- Disabling/Combining/Embedding Assertions
- Controlling Assertions
- Sequences and Sequence Repetition
- Sequences Operators
- Synthesis Assertion Coverage
- Assertion Libraries

Course 3
Language: System Verilog for RTL Design
- Basic System Verilog Features
- Implementing User Logic Intent (combinatorial logic & latches)
- Implementing User Logic Intent (meaning of full and parallel)
- Implementing User Logic Intent (implementing registers)
- Implementing User Logic Intent (implementing state machines)
- Implementing User Logic Intent (wildcard and tri-state logic)
- Advanced System Verilog Features (packed or unpacked array and struct)
- Advanced System Verilog Features (System Verilog interface)
- Advanced System Verilog Features (System Verilog package)
- Achieving High QoR Through Coding

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

Course 4
Language: System Verilog Testbench
- The Device Under Test
- System Verilog Verification Environment
- System Verilog Testbench Language Basics - 1
- System Verilog Testbench Language Basics - 2
- Managing Concurrency in System Verilog
- Object-Oriented Programming: Encapsulation
- Object-Oriented Programming: Randomization
- Voltage Dependent Rule Check
- Object-Oriented Programming: Inheritance
- Inter-Thread Communications
- Functional Coverage
- System Verilog UVM preview

Synopsys Confidential Information
© 2023 Synopsys, Inc.
Language Learning Path

Course 5

Language: System Verilog

Verification using UVM

System Verilog OOP
Inheritance Review

UVM Structural Overview

Modeling Stimulus (UVM Transactions)

Creating Stimulus Sequences (UVM Sequence)

Component Configuration and Factory

TLM Communications

Scoreboard & Coverage

UVM Callback

Advance Sequence/Sequencer

Phasing and Objections

Register Layer Abstraction (RAL)

Summary

Legend

Self-paced Learning

Instructor-Led Training

Downloadable Lab

Cloud-based Lab

Duration in Days

Badge
LynxNXT Learning Path

Course 1

LynxNXT: Foundation

Introduction
Variable Editor
Flow Editor
Execution Monitor
Failure Debug
Exploration
Command Line Interface
Working with FPM

Legend

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

Course 1

PrimeLib: 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

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

## Course 2

### PrimePower: 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

## Course 1
- **PrimeSim: Jumpstart**
  - PrimeSim Static CCK Introduction
  - PrimeSim CCK Built-In Checks
  - PrimeSim CCK Interactive Debugging Commands
  - PrimeSim CCK False Error Pruning
  - Custom Compiler – PrimeSim CCK: Setup and run

## Course 2
- **PrimeSim: Foundation**
  - PrimeSim XA Introduction
  - PrimeSim XA Nelist format Support
  - PrimeSim XA Analyses support
  - Command/Option Usage & Precedence Rules
  - PrimeSim XA Post-Layout Simulation
  - PrimeSim XA Command Line Usage
  - PrimeSim XA Log File Details
  - Accuracy and Speed Trade-off
  - Back-Annotation & XBA
  - Probing in PrimeSim XA
  - PrimeSim XA .ALTER Usage
  - PrimeSim XA .DATA Usage

## Course 3
- **PrimeSim: Advance**
  - CCK Advanced ERC and ESD Checks
  - CCK Propagation Engine – XPL and Analog Propagation
  - CCK Custom Programmable Checks
  - Custom Check Assertion
  - GUI: Cross-Probe, Filtering, Waiver, Grouping, export

## Course 4
- **PrimeSim: Advance**
  - Interactive Mode
  - Distributed Processing
  - Monte-Carlo (MC)
  - MOSRA
  - Aging and Self-Heating

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

**Course 1**
- **PrimeTime: 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**
- **PrimeTime: 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**
- **PrimeTime: 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**
- **PrimeTime: 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

© 2023 Synopsys, Inc.
# Recommend Learning Journey: Physical Designer

<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>
</table>

**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:  
- **Fusion Compiler: UPF Fundamental**  
  - Introduction UPF  
  - Power Domains  
  - Power Strategies  
  - Supply Network  
  - Power States  
  - Fusion Compiler and UPF

Course 7:  
- **Reference Methodology: 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

Reference Methodology: Jumpstart

- Introduction & Overview
- Organization & Structure
- Running RM
- Demo

Legend

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

Synopsys Confidential Information
RTL Architect Learning Path

Course 1: RTL Architect: 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: RTL Architect: 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  Course 2

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

StarRC: 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
SaberRD Learning Path
Course 1

SanerRD: Foundation
Training Series

Timing Domain Analysis
Schematic Capture & Parts Library
Operating Points & Small Signal Frequency Analysis
Test Automation
Design Optimization
Introduction to Modeling
Import SPICE Models
Modeling with Table Look-Up
Modeling with StateAMS
Robust Design & Sensitivity Analysis
Monte Carlo & Pareto
Worst-Case Analysis
Fault Analysis
Stress Analysis

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

Course 1

Synplify: 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

TestMAX: 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
- TestMAX: 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
- TestMAX: ATPG
  - 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
  - Path Delay Testing
  - Power Aware ATPT
  - Conclusion

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

Course 1

TestMAX FuSa: Jumpstart

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

Legend

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

Course 1

TestMAX Manager: 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

Synopsys Confidential Information
TestMAX SMS Learning Path

Course 1

TestMAX-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

### TestMAX XLBIST: 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**

© 2023 Synopsys, Inc.
Tweaker Learning Path

Course 1

Tweaker: 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
VC Formal Learning Path

Course 1

VC Formal: Foundation

Introduction
Formal Verification Methodology
SVA for Formal Verification
VC Formal Basic
VC Formal Navigator

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: Advanced
  - 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