SOPC WORLD 2002

SOPC WORLD 2002

Techniques of Maximizing FPGA Design Performance

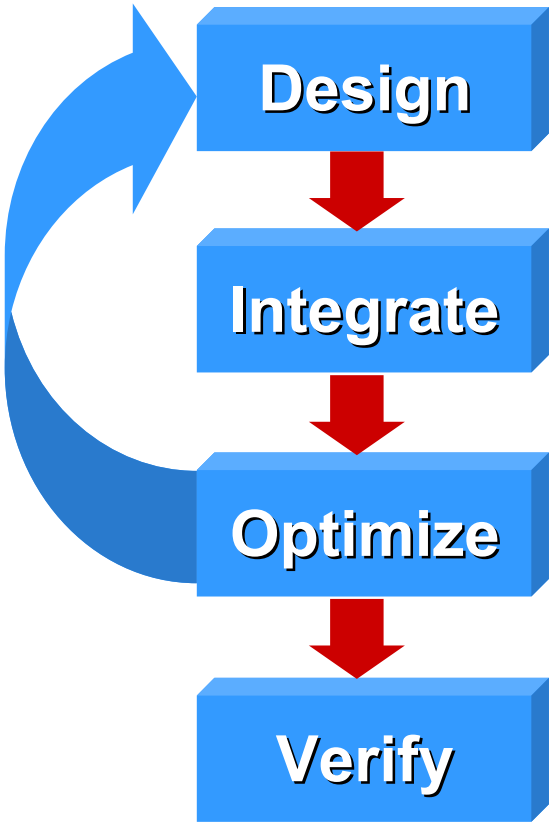
ALTERA.

Hierarchical Block-Based Design

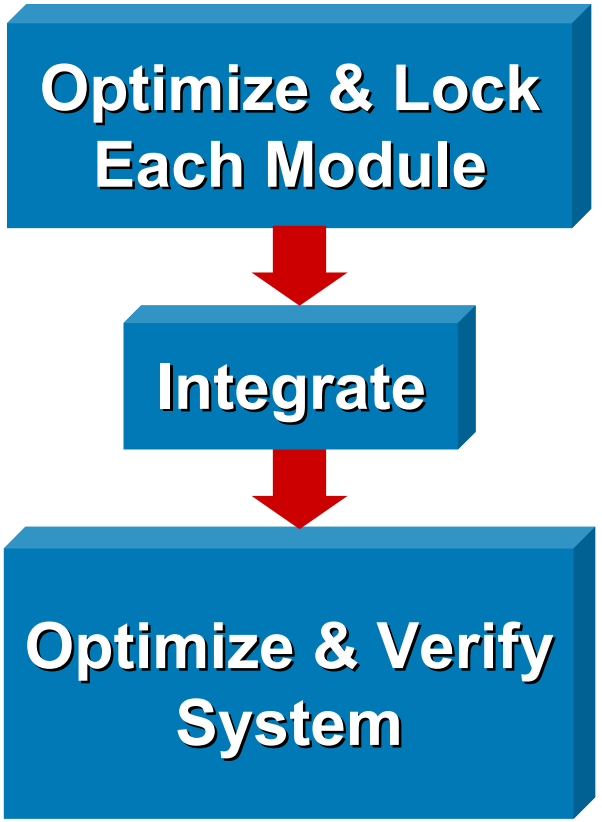
- As FPGA Designs Become Larger, New Techniques Needed to Reduce Design Cycle Time
- Hierarchical Block-Based Design Flow Referred to as LogicLock™ Design Flow
  - Facilitates Team-Based Design
  - Allows Easier Module Reuse
  - Used for Incremental Recompiles

Design Flows

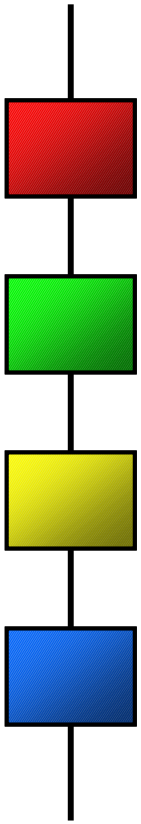
Old Design Flow



LogicLock Design Flow



LogicLock Design Flow



Partition Design

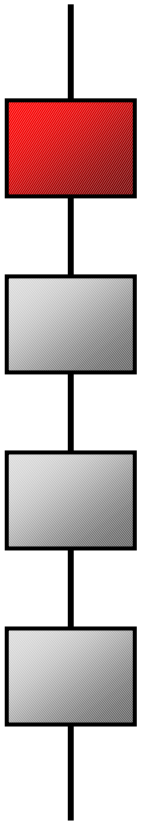
Synthesize Modules

Optimize Modules

Integrate Modules

LogicLock™

LogicLock Design Flow



Partition Design

Synthesize Modules

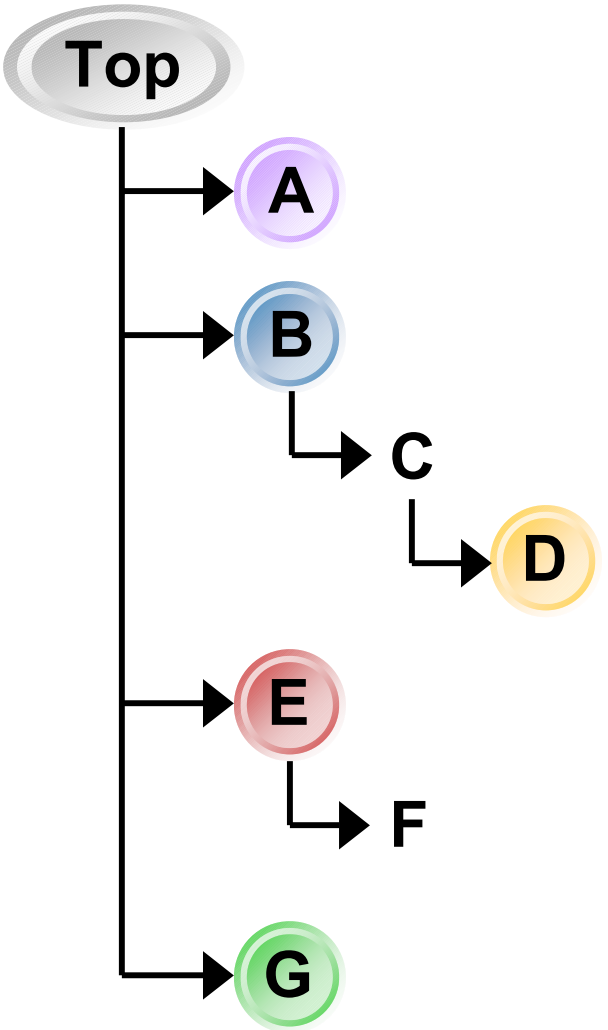
Optimize Modules

Integrate Modules

LogicLock™

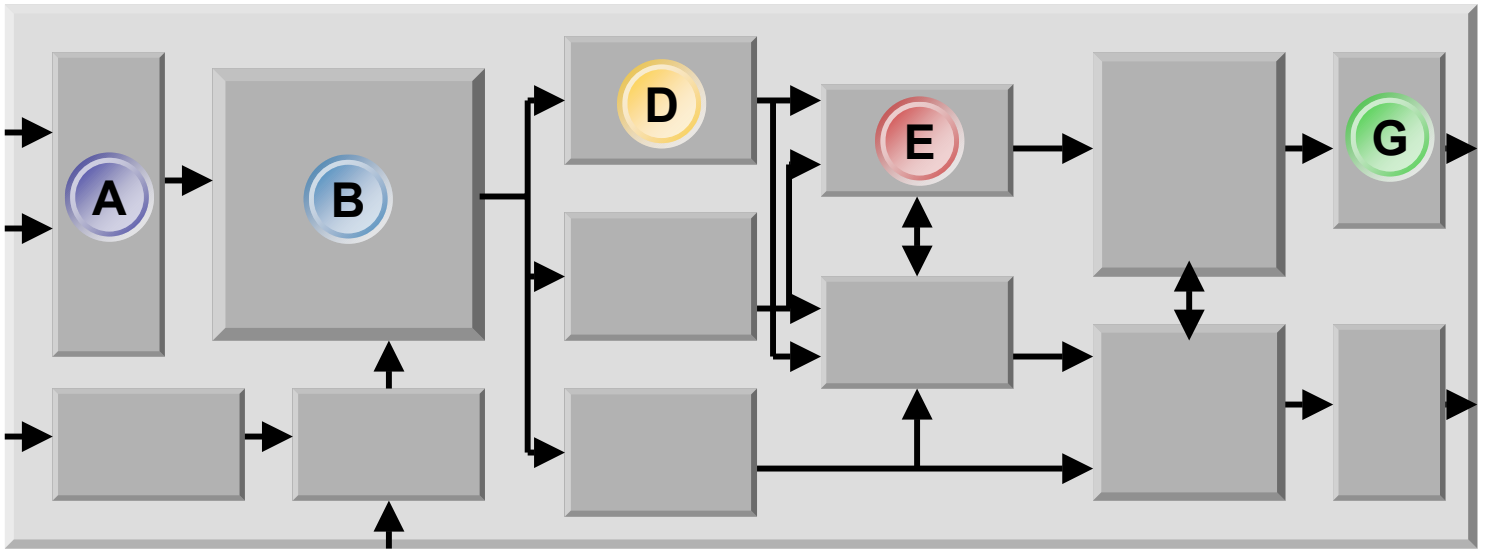
Design Partitioning

- Need to Partition Design into Modules
- For Optimal Results, Designs Must Be Partitioned Correctly
- Should Be Done Early in Design Cycle



Defining Partitions

- Start with Design Block Diagram
- Maintain Offchip I/O Interface Modules
- Identify Different Clock Domains
- Divide Based on Functionality



Defining Partitions

- If Using Team-Based Design, Divide by Engineer
  - Generally Corresponds to Functionality
- Balance Size of Partitions
  - To Ensure that Small Modules Are Optimized with Other Modules
  - Requires Estimating Size of Modules
    - Designer's Expertise
    - Previous Similar Designs Done by Engineer
    - Calculations on Functional Data

Partition Boundaries

- Register Input & Output Boundaries
  - May Be Larger Routing Delay Between Partitions Depending on Location on FPGA
  - Register-to-Register Paths with Large Routing & Combinatorial Delay Will Hurt f_{MAX}

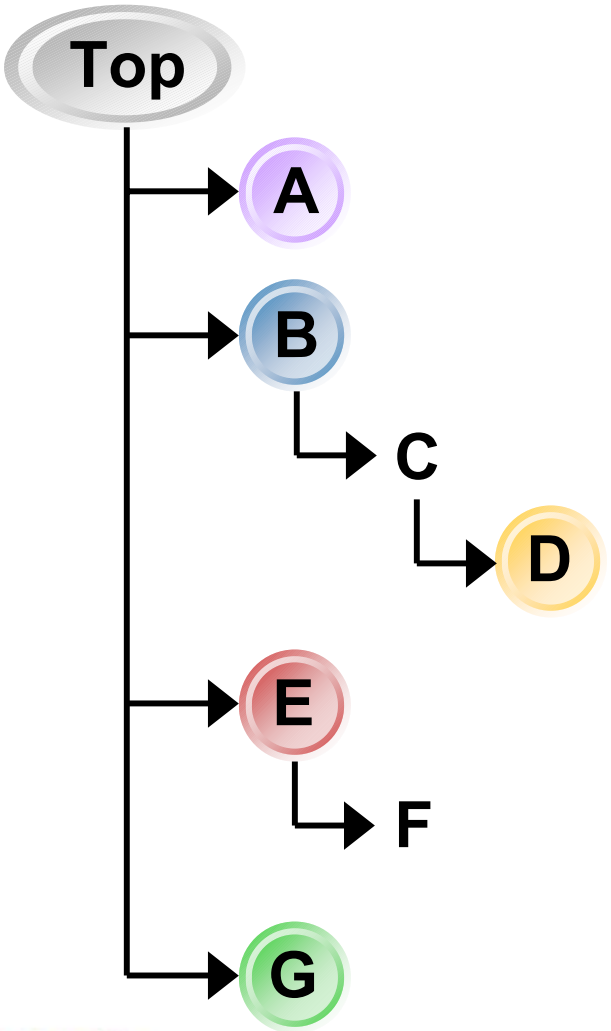
Partition Boundaries

- Minimize the Data Path between Modules
  - Clearly Define Data Path & Distribution of Common Data
  - Best to Have Partitions after Contracting vs. Expanding Math Functions

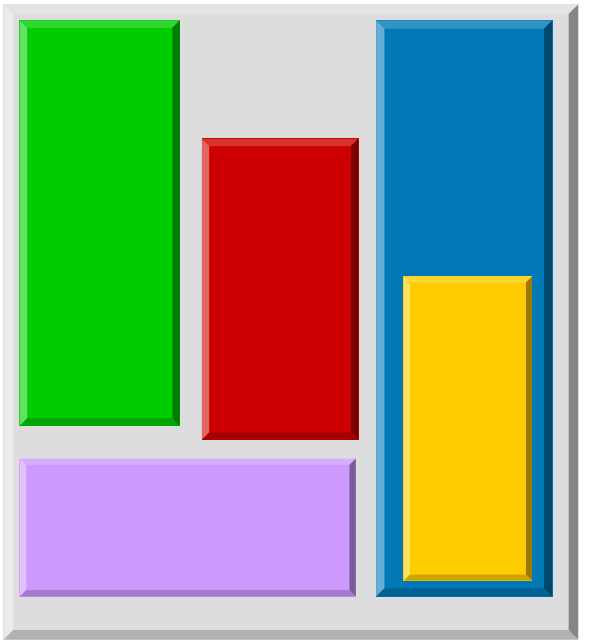
Partition Boundaries

- Resources with Flexible Locations Should Be At Partition Edges
  - DSP Blocks or RAM Blocks Often Have Fixed Locations & Are Less Flexible
  - Flexible Resources (Logic Elements) Can Reduce Routing Delay as They Can Be Placed at Physical Edge of Partition If Necessary

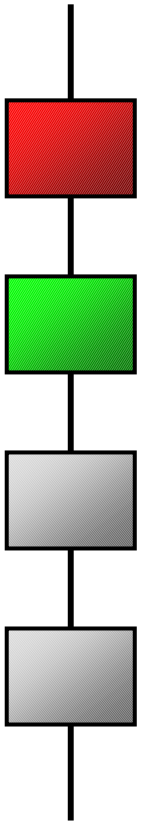
Initial Floorplanning



- Designers Can Perform Initial Floorplanning Based on Partition Estimates



LogicLock Design Flow



Partition Design

Synthesize Modules

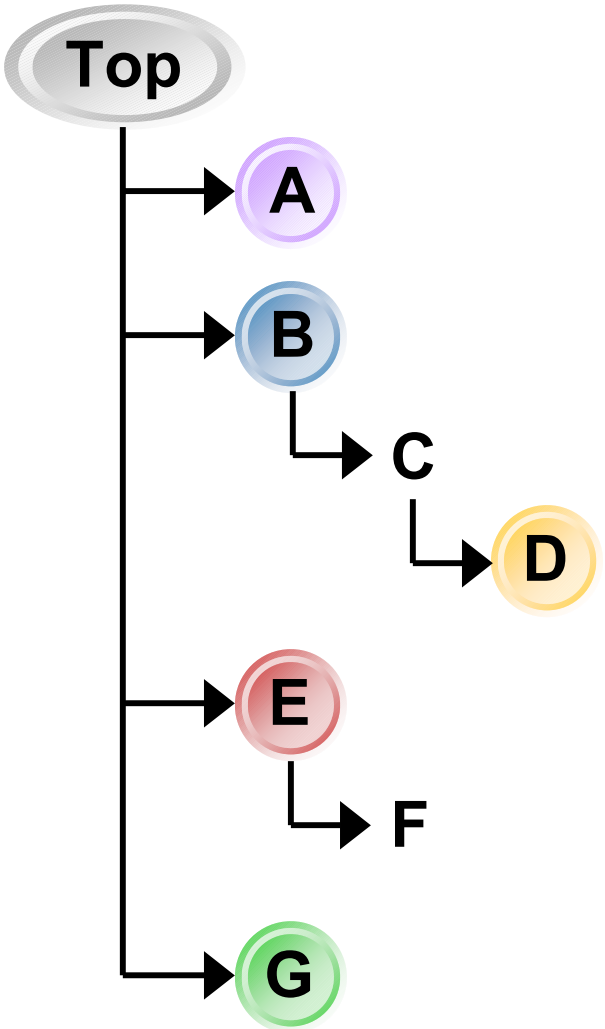
Optimize Modules

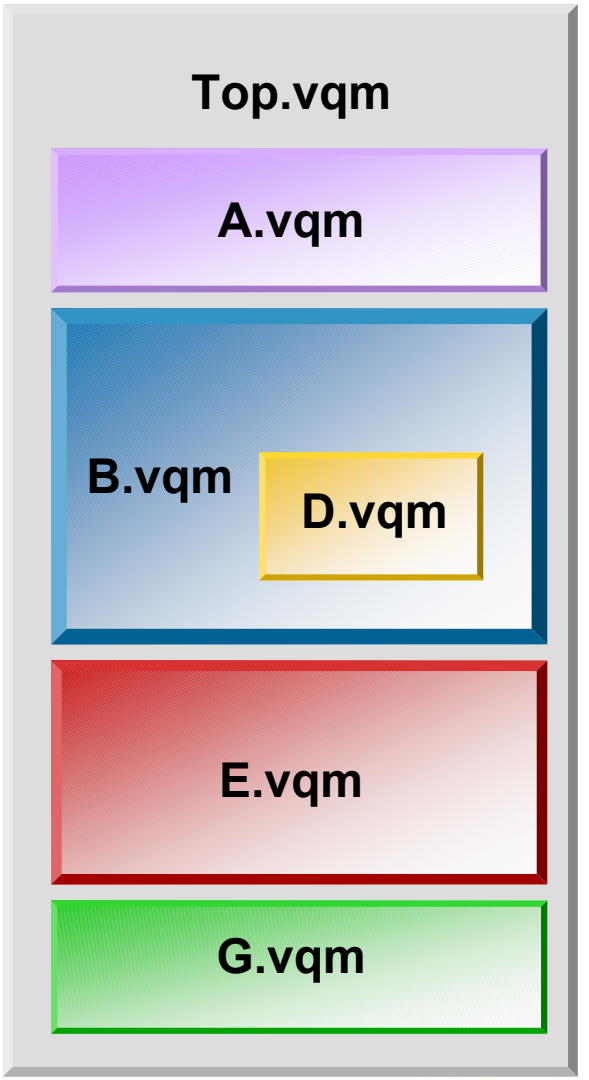
Integrate Modules

Synthesize Modules

- Each Partition Will Have Separate Atom Netlist (EDF or VQM file)
  - Atom Netlist Defines Logic in Terms of FPGA Primitives
- Can Choose to Have Separate Synthesis Projects for Partitions or Use LogicLock Flow Developed by Major Synthesis Tools

Synthesize Modules





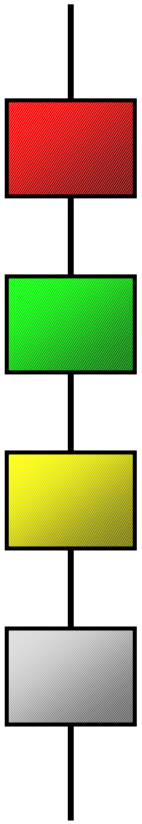
Synthesis Tools

- LeonardoSpectrum™ Tool
  - Initial Project Can Be Split into Separate EDIF Files
  - TCL Flow for Incremental Changes
- Synplify®
  - MultiPoint difference-based incremental synthesis produces separate VQM files
- FPGA Compiler II
  - BLIS Flow Splits Initial Projects & Regenerates EDIF Files for Incremental File Changes
- Quartus® II
  - Separate Projects for Each Partition to Generate VQM Files

Preserving Changes

- Whole Design Characterized Somewhere in Atom Netlist
  - Preserves Nodes & Node Names
- Can Generate a Quartus II VQM File to Preserve Changes
  - If Altera Megafunctions Used
    - Megafunctions Only Synthesized in Quartus II
  - If Quartus II Netlist Optimization Options Used

LogicLock Design Flow



Partition Design

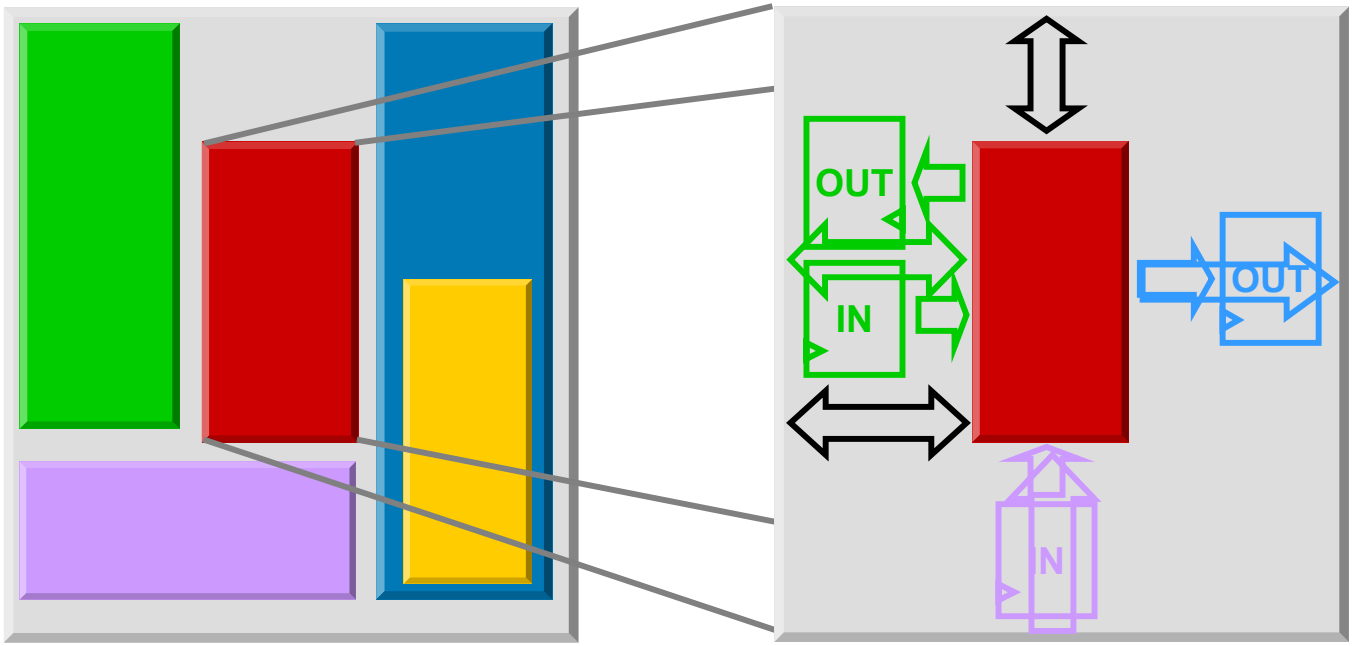
Synthesize Modules

Optimize Modules

Integrate Modules

Optimize Independent Modules

- Use the Quartus II Software to Optimize Each Module Independently
- Quartus II Has Virtual I/O Feature to Allowing Accurate Timing Analysis

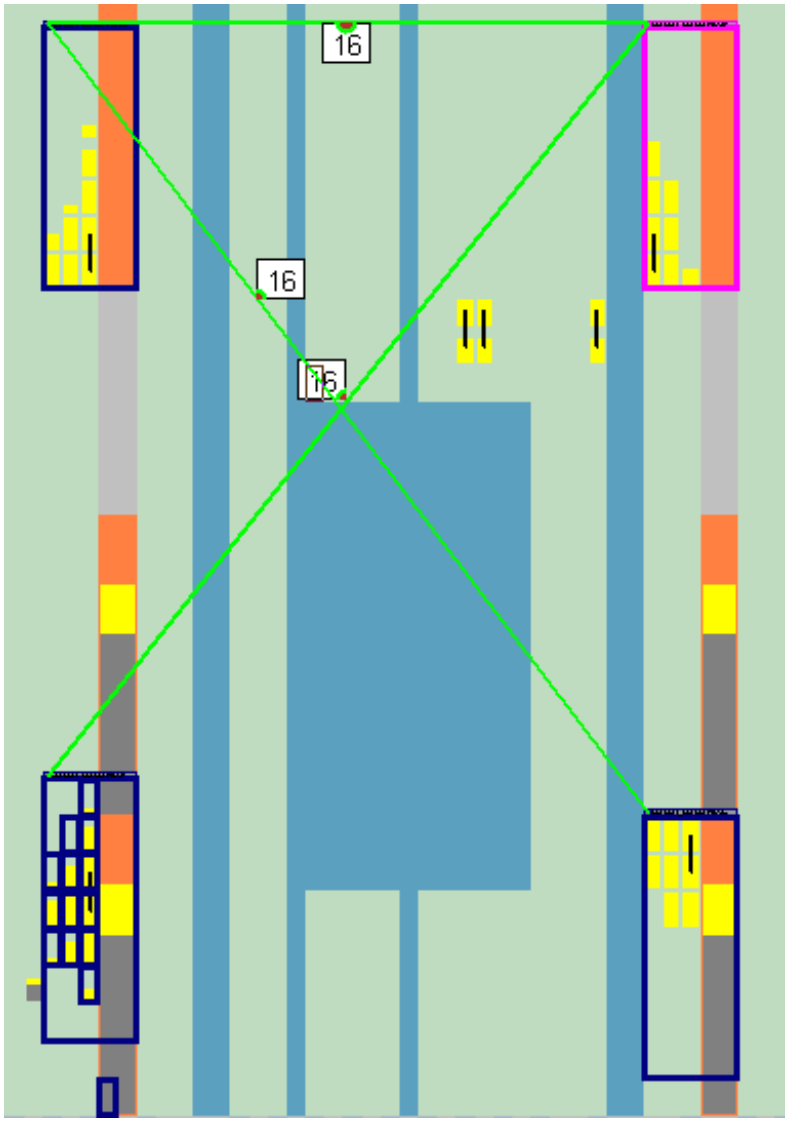


Optimize Independent Modules

- Create Separate Quartus II Projects for Each Module
- Perform Design Analysis
  - Quartus II Timing Analysis
  - New Timing Closure Floorplan
- Optimize If Necessary

Design Analysis

- Timing Closure Floorplan Provides Number of Features for Graphical Analysis
  - Viewing Critical Paths
  - Connectivity between Modules
  - Physical Delay Estimates
- Closely Integrated with Static Timing Analysis Results

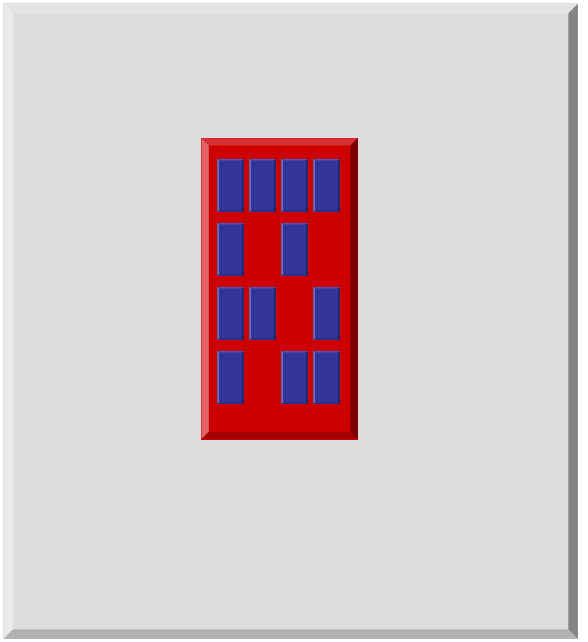


Optimization Methods

- Optimize Design If Necessary Using Quartus II Options
- All Optimization Methods Available for Lower-Level Modules
  - Netlist Optimizations
    - Options to Optimize Design after Synthesis & before Place & Route
    - Can Be Used Regardless of Synthesis Tool
  - LogicLock Assignments
  - Location Assignments

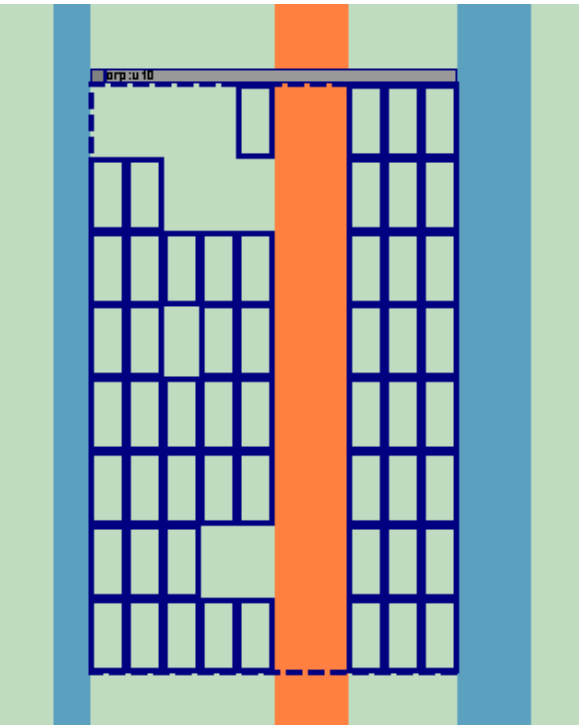
Module Placement Preservation

- Lock Down Placement of Module Using LogicLock Regions
- Performance Preserved in Top Level Design
- Design Information Stored in Atom Netlist
  - VQM or EDIF File
- Placement Information Stored in Quartus II Constraint File
  - ESF File

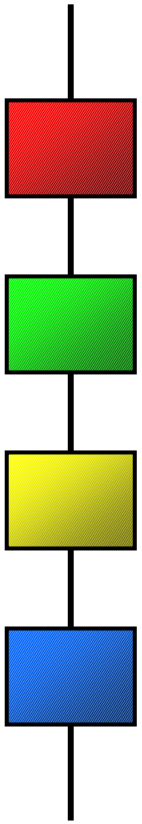


LogicLock Region

- Contiguous, Rectangular Block of Device Resources
- Design Nodes or Entities Assigned to LogicLock Regions
- LogicLock Regions
  - Can Be Hierarchical
  - Do Not Have to Have Fixed Size or Location
  - Can Maintain Relative Placement of Nodes within Them



LogicLock Design Flow



Partition Design

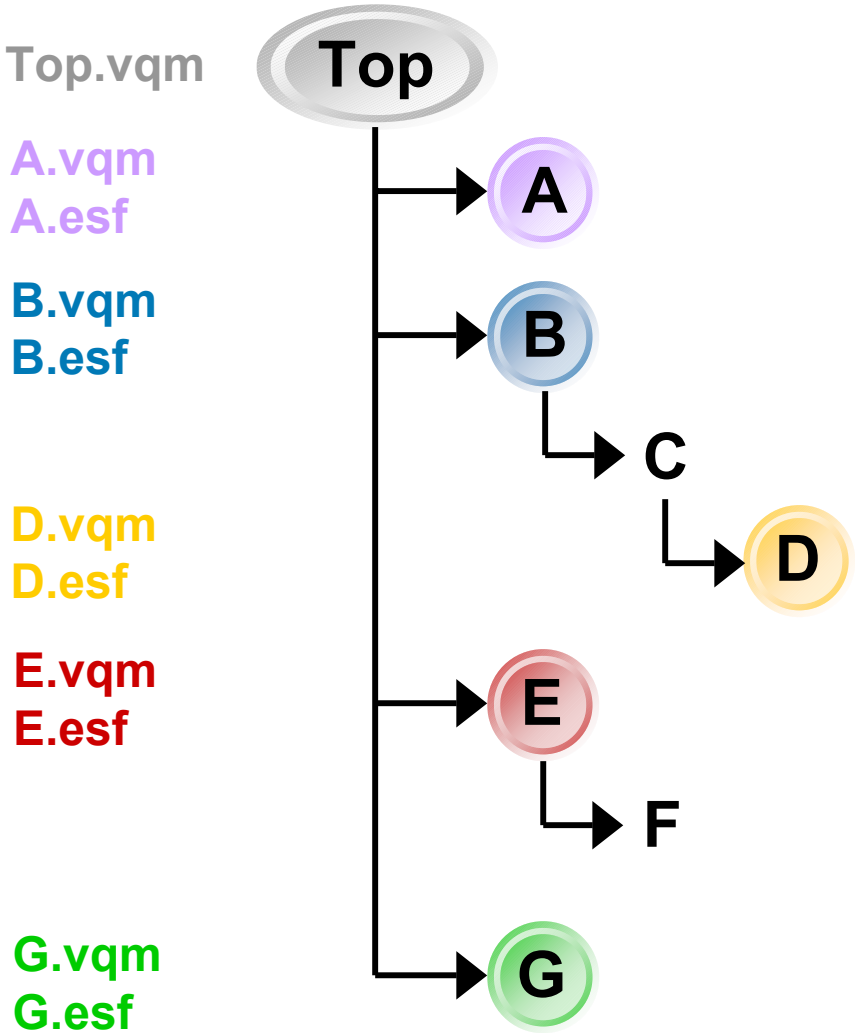
Synthesize Modules

Optimize Modules

Integrate Modules

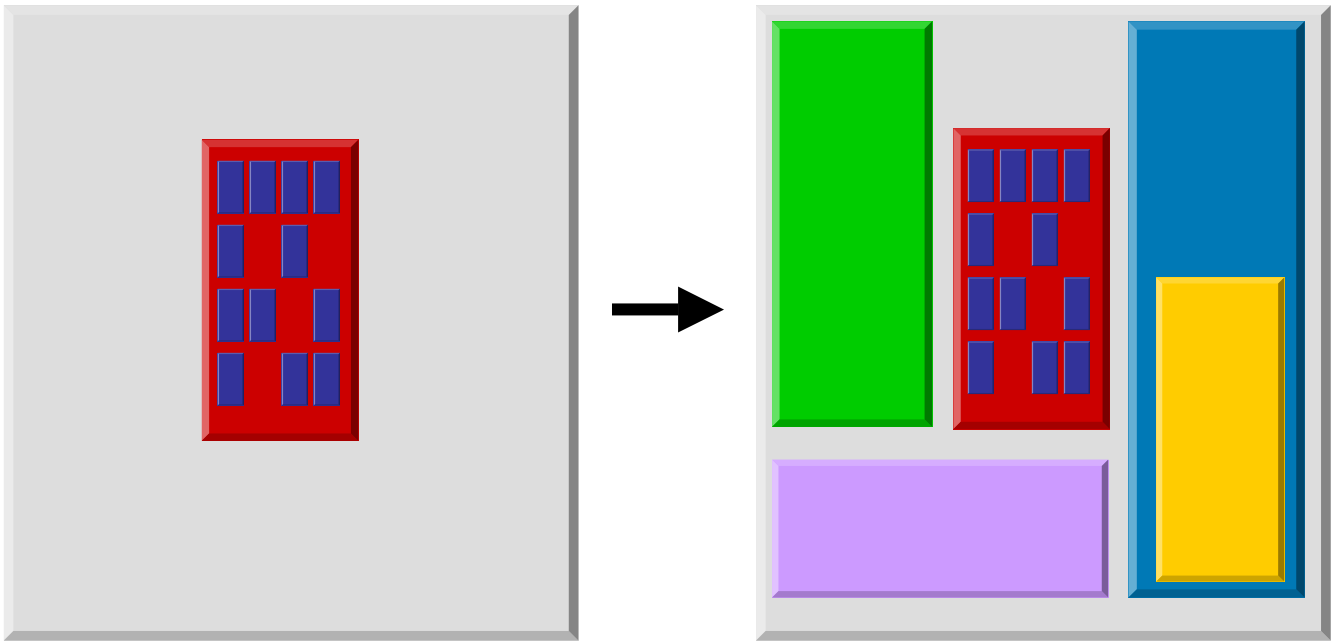
Integrate Modules

- Import Each Module into Top Level
- Requires Files for Each Module
  - Atom Netlist (EDIF or VQM)
  - Placement Constraint File (ESF File)



Integrate Modules

- Performance of Each Module Preserved because Relative Locations of Nodes Maintained



Summary

- Hierarchical Block-Based Design Effective Technique to Reduce Design Cycle Time
- LogicLock Design Flow Allows Method to Preserve Performance of Design Modules