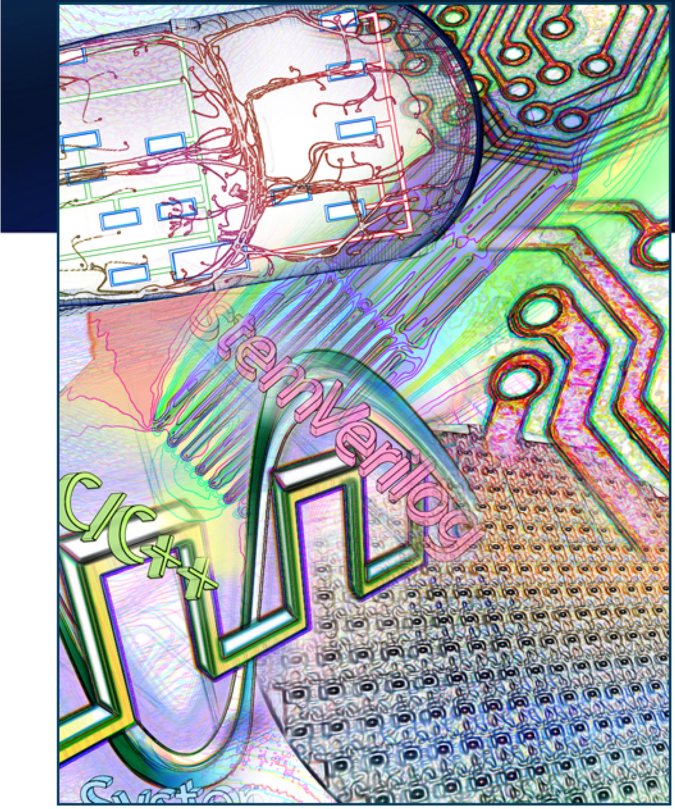
Mentor’s FPGA Implementation Flow for Atmel



FPGA Synthesis & Equivalence Checking

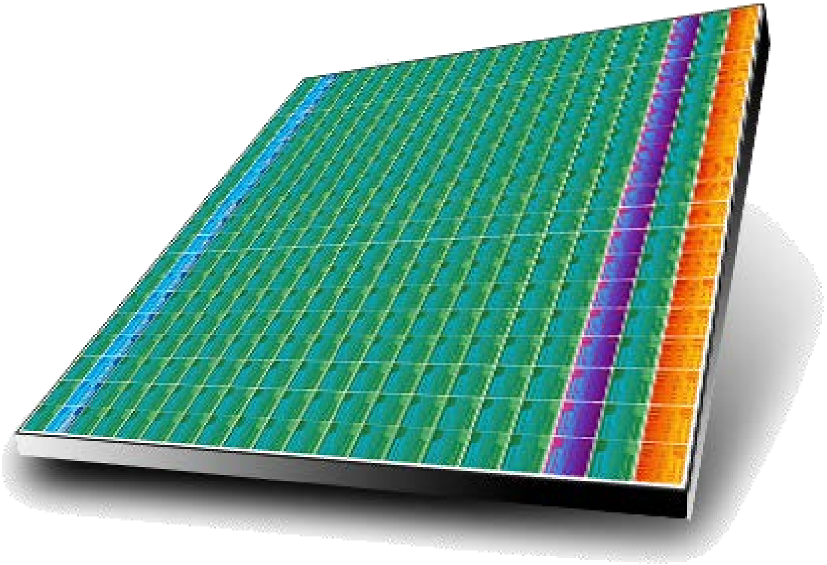
November 2012

Olivier Takeznount

Applications Engineer Europe

Precision Synthesis for FPGA Implementation

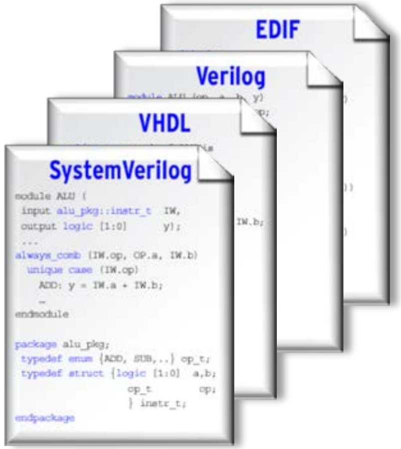
- Design
  - Leading mixed language support
  - Precise-IP Platform
- Synthesize & Optimize
  - Out-of-the-box Quality of Results
- Achieve Design Closure
  - Award-winning analysis & debug
- Atmel-Specific Options & Enhancements



- Formal Verification using Equivalence Checking

Leading Mixed Language Support

- Verilog, SystemVerilog, VHDL 2008, EDIF
- Support for mixed language design input
- Synopsys Design Constraint (SDC) Support
  - — ASIC Industry standard for timing constraint



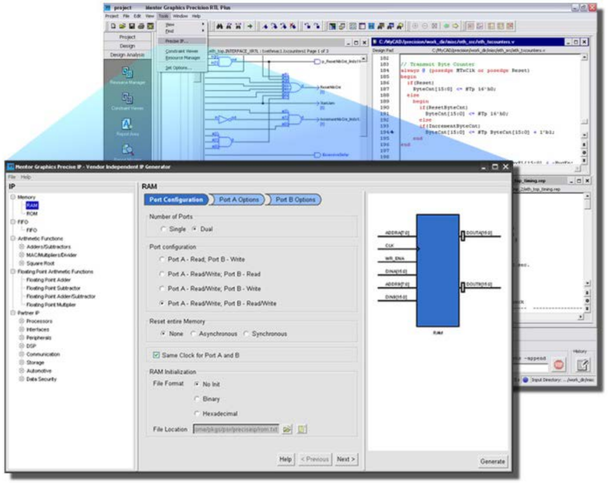
```
module ALU (
 input alu_pkg::instr_t  IW,
 output logic [1:0]      y);
 ...
always_comb (IW.op, OP.a, IW.b)
  unique case (IW.op)
    ADD: y = IW.a + IW.b;
    …
endmodule

package alu_pkg;
 typedef enum {ADD, SUB,..} op_t;
 typedef struct {logic [1:0]  a,b;
                 op_t         op;
                 } instr_t;
endpackage
```

Precise-IP™, IP Encryption: Vendor Independent IP Platform

IP

- Precise-IP
  - — Vendor Independent IP Platform
  - — Expanded library of configurable cores
  - — Re-target IP to different FPGAs
  - — Out-of-the-box Quality of Results
  - — Over 70 certified cores from leading IP vendors
- Precise-Encrypt
  - — Vendor independent encryption
  - — Synthesize encrypted IP (IEEE P1735 draft)
  - — Share IP securely with teams, vendors, partners

Precise-IP™ Platform



Mentor Graphics Precise IP - Vendor Independent IP Generator

RAM

Port Configuration | Port A Options | Port B Options

Number of Ports: ○ Single ● Dual

Port configuration:
○ Port A - Read; Port B - Write
○ Port A - Read/Write; Port B - Read
○ Port A - Read/Write; Port B - Write
● Port A - Read/Write; Port B - Read/Write

Reset entire Memory: ● None ○ Asynchronous ○ Synchronous

☑ Same Clock for Port A and B

RAM Initialization
File Format: ● No Init ○ Binary ○ Hexadecimal
File Location: /ome/pkgs/psr/preciseip/rom.txt

IP Vendors

ARM

AEROFLEX GAISLER

IPextreme

CAST

Inno-Logic ASIC.FPGA.Embedded

ALIATHON TECHNOLOGY

EnSilica your trusted IC design partner

OptNgn Your HardwareCode Place

LogicDesignSolutions

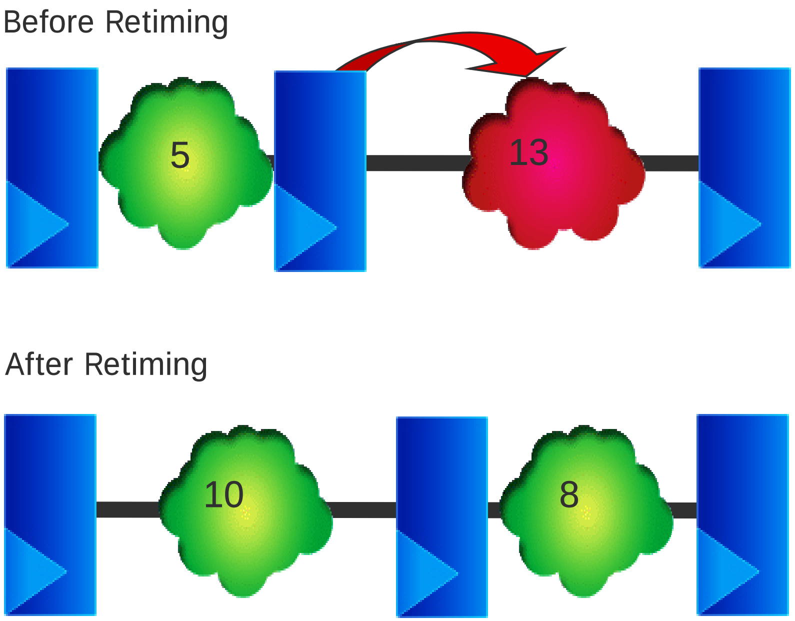
Eureka Technology

HELION

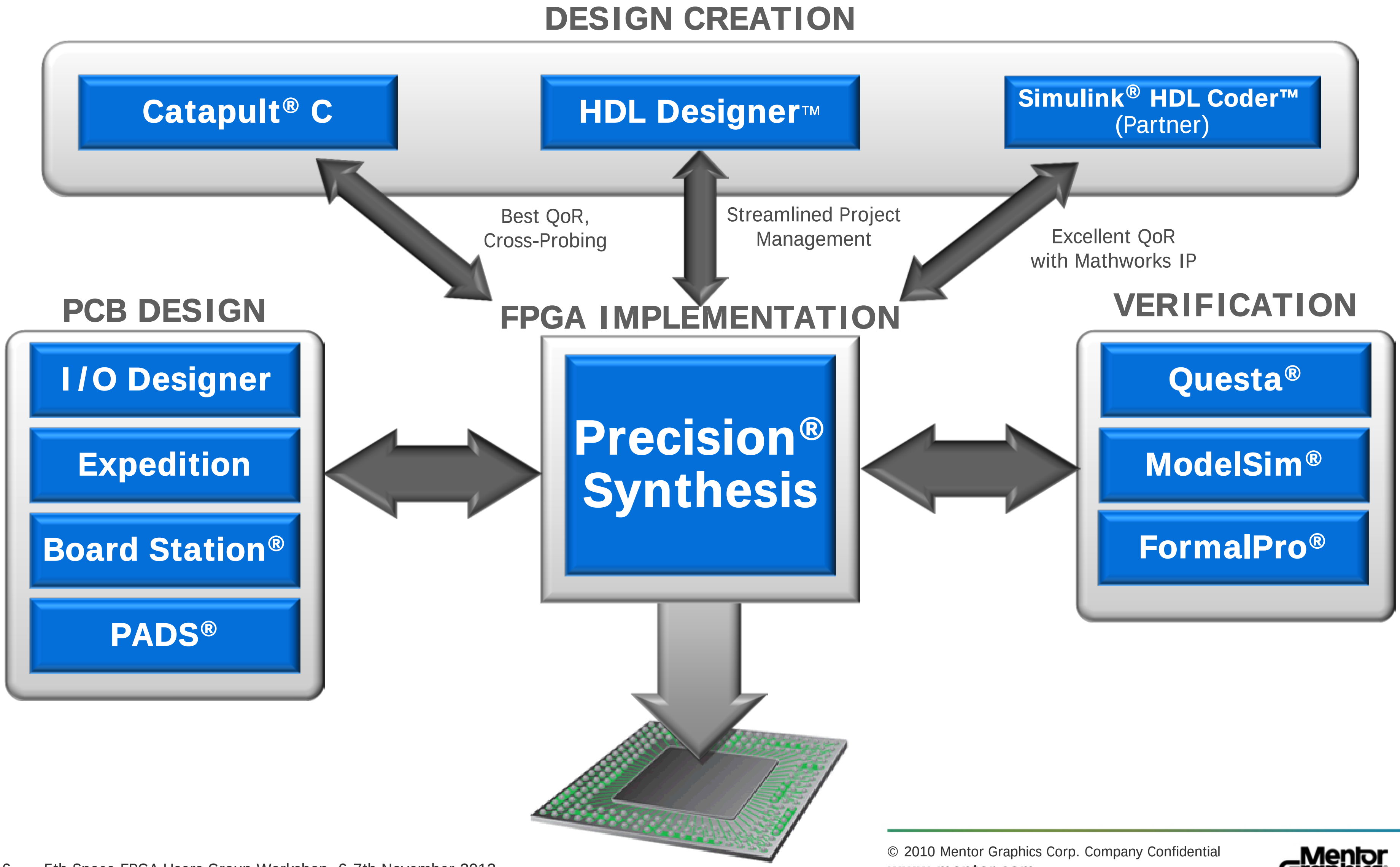
Out-of-the-Box Quality of Results (RTL Synthesis)

- Advanced technology-independent inference
  - — Memories, DSP elements, Operators, Shifters
- Advanced optimizations
  - — Retiming
  - — Data-Path Optimization
  - — Resource sharing
  - — Timing Driven Optimization

Retiming



Mentor's Integrated, FPGA Vendor Independent Tool Flow



DESIGN CREATION

Catapult® C

HDL Designer™

Simulink® HDL Coder™ (Partner)

Best QoR, Cross-Probing

Streamlined Project Management

Excellent QoR with Mathworks IP

PCB DESIGN

I/O Designer

Expedition

Board Station®

PADS®

FPGA IMPLEMENTATION

Precision® Synthesis

VERIFICATION

Questa®

ModelSim®

FormalPro®

Atmel-specific options in 2010a update 2

- Default behavior is that Precision maps to LPM macros and doesn't map to FGEN2 cells. So, if that is the behavior desired by the user, then nothing should be done.

- Precision can map to FGEN2 cells by using the following command:
  - — setup_design -var at280k_new_flow=true
  - — The default value in 2012a update2 is false

- To disable inference of LPM macros and map operators to logic resources, the following 2 commands need to be specified:
  - — setup_design -var map_atmel_lpm_to_tech_cell=true
  - — setup_design -var at280k_new_flow_resolve=true
  - — The default values for these variables in 2010a update 2 is false

  Note:
  - — setup_design -var at280k_new_flow_without_lpm=true is a superset variable that enables the 2 variables above and enables mapping to FGEN2 cells

Atmel-specific options in 2010a update 3

- All the options above are merged into one command:
  - setup_design -atmel_map_options {}

- The curly brackets are required as the values that can be provided to the option could be one or more of the following 4 pre-defined values:
  - lpm: map operators to LPMs
  - no_lpm: disable mapping operators to LPMs and map them to logic cells
  - single_output_macros: map logic to FGEN1* primitives
  - dual_output_macros: map logic to FGEN1* and FGEN2* primitives

- Also, note that multiple values can be specified using the space character as a separator like:
  - setup_design –atmel_map_options {lpm dual_output_macros} or
  - setup_design –atmel_map_options {no_lpm single_output_macros}

Atmel-specific options in 2010a update 3 (cont'd)

- The default behavior in 2010a update 3 is maintained from 2010a update 2. For example:
  - — setup_design –atmel_map_options {lpm single_output_macros}
- If the user specifies 2 conflicting options in the list, then the last one prevails. For example:
  - — setup_design –atmel_map_options {lpm no_lpm single_output_macros} => in this case, **no_lpm** prevails along with **single_output_macros**

2010a update 3 Enhancements

- Write functional VHDL simulation netlist for FGEN2 instances
- Inference of LPM_MUX
- Inference of carry select adder LPM (LPM_ADD_SUB_CS) for adders with 24 bits or more
- Inference of RSBUF reset buffer instead of IBUF
- Fix LPM_ROM cut-off to address width of 10 instead of 9
- Enable register retiming for ATF280E
- FA and MULT introduction

What’s New in 2012b[1]

[1] 2012b released in October

- New Family/Device Support
  - ATFS450 (All ATF280E options are available for ATFS450)
  - ATF280E part/package - enhancement support

- Enhanced FGEN Mapping
  - Mapping of FGEN*R, FGEN*RT
  - Improves area QoR

- Enhanced MGEN Mapping
  - Functional extraction of AND-gate patterns
  - Mapping of MGEN*R, MGEN*RT
  - Improves area QoR

- Enable register retiming for ATF280F & ATFS450

What’s New in 2012b (continued)

- Inference of LPM_MUX
- Clock buffer inference for fanout > 1
- IOB registers not used by default
- Area report enhancement for FGEN*R, MGEN*R & MULT
- Write functional VHDL post-synthesis simulation netlist
  - Currently assign statements are used => will be replaced with appropriate technology-specific primitives

What’s New in 2012b (continued)

- Hierarchy Flattening
  - ON by default for improved P&R flow
  - Use the following new option to turn it OFF:
    - setup_design -atmel_map_options {no_flatten_hierarchy}

- Consolidate & Simplify Atmel Mapping Options
  - Specify mapping options using only the following command:
    - setup_design -atmel_map_options {}

  - Precision will give an error message when old variable options are specified using the following command:
    - setup_design **-var** <option_name>

Overview of Atmel-specific options in 2012b

- Specify mapping options using the following command:
  - setup_design -atmel_map_options {}

- The curly brackets are required as the values that can be provided to the option could be one or more of the following 5 values:
  - lpm: map operators to LPMs
  - no_lpm: disable mapping operators to LPMs and map them to logic cells
  - single_output_macros: map logic to FGEN1* primitives
  - dual_output_macros: map logic to FGEN1*, FGEN2* and MGEN* primitives
  - no_flatten_hierarchy: does not flatten user hierarchy, uses auto mode

- Multiple values can be specified using the space character as a separator:
  - setup_design –atmel_map_options {lpm dual_output_macros} or
  - setup_design –atmel_map_options {no_lpm single_output_macros}

Formal Verification Using Equivalence Checking

- Why use Logic Equivalence Check ?
- Where does Equivalence Check fit in ?
- Inputs/Outputs & Results
- FPGA Support
- Summary

Why use LEC ?

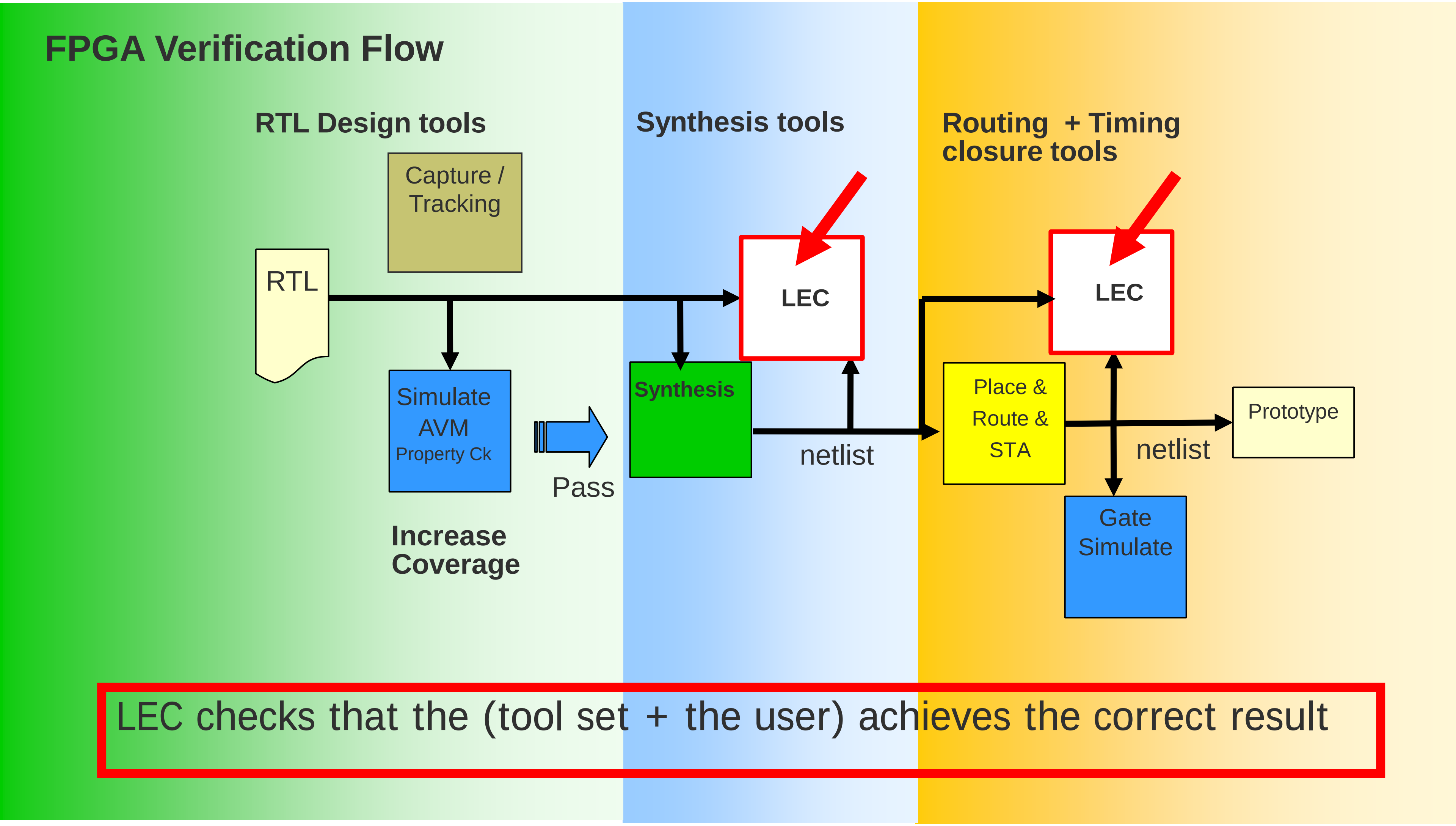
- LEC provides higher functional coverage in much faster time

- LEC static analysis proves that all functions of the reference design are implemented in the netlist

  — Detects functional differences
  — Detects unused and extra logic
  — Detects floating and shorted nets

- Static Timing reports can confirm correct netlist timing
- LEC + Static Timing meet the objectives of Gate simulation
- LEC transfers RTL functional coverage to a netlist

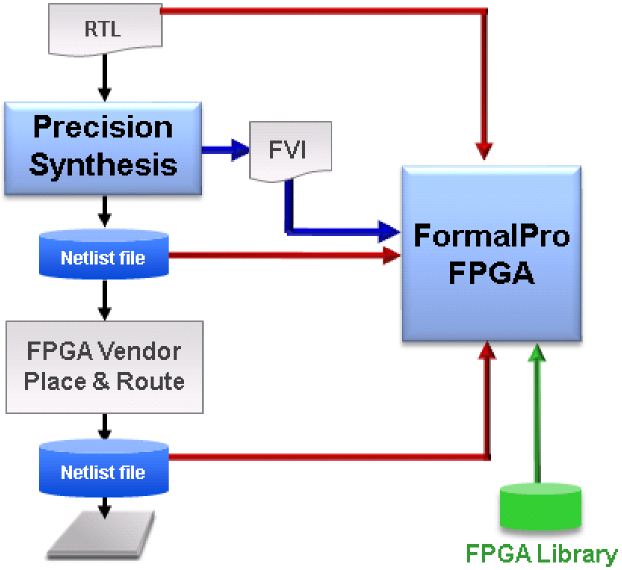
Where does Equivalence Check fit in ?



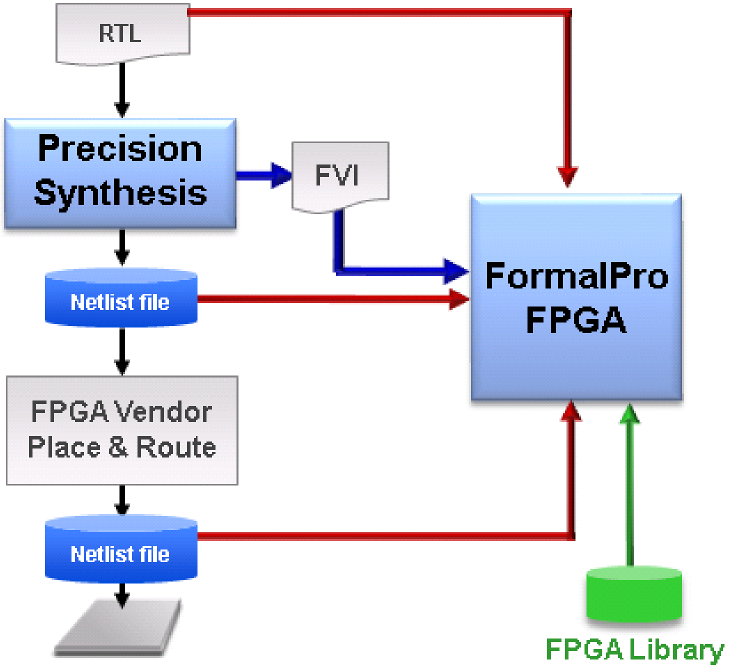
LEC checks that the (tool set + the user) achieves the correct result

Inputs/Outputs

- LEC inputs
  - HDL or netlist format
    - Must be synthesizable to logic
    - Verilog 1995/2001, VHDL 87/93, SystemVerilog
    - Cannot be a protected IP
    - Any combination for A/B comparisons
      - RTL<->RTL applications
  - Guide File
    - Precision, Synplify_pro, Design_Compiler, user
  - Technology Library
    - FPGA FV libraries (not simulation-lib)
      - Actel, Altera, Atmel, Xilinx
- Outputs
  - `EQUIVALENT, DIFFERENT, No Differences Found(exceptions)`
  - Compile reports, Match reports, Solve reports
    - Matched A/B pairs which form targets
    - Ignorable instances (unused logic)
    - Lists of removed instances (non-qualified targets)
  - Debug tools to locate errors
  - Schematics



Simple Results



```
Comparison Summary
==================
  Total number of comparison points:          1889
  -----------------------------------------------------

  Number of Equivalent comparison points:     1889
  -----------------------------------------------------

  Number of Different comparison points:      0
  -----------------------------------------------------

  Number of Removed comparison points:        0
  -----------------------------------------------------

  Number of Unsolved comparison points:       0
  -----------------------------------------------------


=============================================
-         Designs are EQUIVALENT            -
=============================================
```

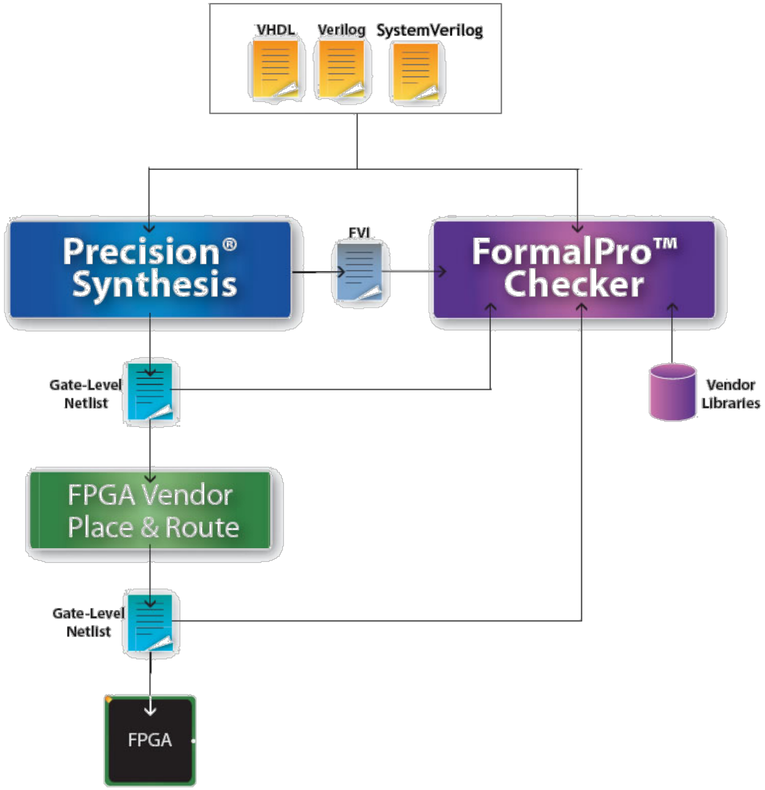
FPGA Support

- The Major FPGA vendors made libraries for Formal Verification
  - These are not the same as simulation libraries because they must be synthesizable
  - Actel, Altera, Atmel, Xilinx (All devices in production)
- The Synthesis tool Must prepare a “guide” file
  - The alternative is too hard for users
- Synthesis
  - Precision (Mentor)
  - Synplify_pro (Synopsys)
- Routing
  - Actel
  - Altera
  - Atmel
  - Xilinx

Automation in the Precision - FormalPro Flow

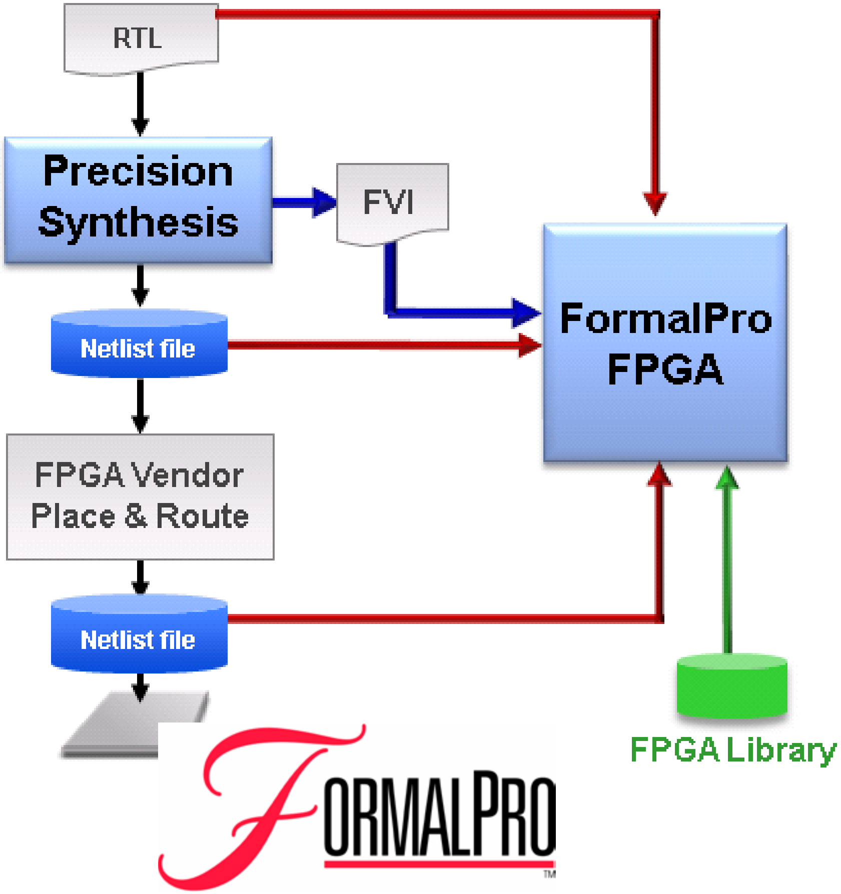
- Precision provides guide/setup file
  to FormalPro (FVI file)
- Supported Precision optimizations
  - Merged registers
  - Duplicated registers
  - Inferred counters
  - Eliminated registers
  - FSMs
- All “guidance” is either self-checking or independently fabricated

  - New state table for FSM



FormalPro LEC Summary

- FPGA synthesis is complex and needs verification
- FormalPro LEC can be applied efficiently to FPGAs with help from synthesis
- LEC provides Functional test coverage of netlist Logic and connections
- LEC transfers RTL coverage to the netlist by equivalence
- Design must pass STA to verify timing
- FormalPro LEC provides verbose records of what is and is not tested



Formal Equivalence Checking

FormalPro 2011.3 (release November 2012)

- Multi-core processing : Makes use of multi-core CPUs by default

  (**-mp 2** is now the default to enable faster results)
- Introducing Windows XP & Windows 7 platform support : Supported with a 2011,3 installer. This build accepts existing Linux license files/keys. True PC support for PC based FPGA developers.
- Solaris 10 no longer supported : allows focus on optimizing Linux and Windows platforms
- Low disk space warning : New warning message if disk space is less than 2Gb at the start of Compile, Match or Solve (provides a clue if subsequent execution fails for disk access reasons)
- ROADMAP
  - Continue improving FPGA memory support
  - Work on retiming in R&D

Mentor Graphics®

www.mentor.com