



EDAC Briefing: Encryption in EDA Software

Pamela Parrish ~ EDA Consortium

Larry Disenhof ~ Cadence Design Systems

Douge Martin ~ Mentor Graphics

Erik Oliver ~ Synopsys

Roz Thomsen ~ Thomsen and Burke LLP

This white paper publicly available at:

http://www.edac.org/resources_export.jsp



What Is EDA?

Electronic – anything electronic—from computer chips, cellular phones, pacemakers, controls for automobiles and satellites to the servers, routers and switches that run the Internet. *Everything* made by the nearly \$1 trillion electronics industry results from designers using EDA tools and services.

Design – the part of the production cycle where creativity, new ideas, ingenuity and inspiration come to the fore. This is also where designers try to model the behavior of their designs and analyze the complex interactions of millions of constituent parts in their designs to ensure completeness, correctness and manufacturability of the final product. Why? Because it is impossibly difficult, expensive and time consuming to "build it first and fix it later."

Automation – imagine the difference between designing a small house versus designing a mile-high skyscraper. For the skyscraper you need to design sophisticated structural, electrical, plumbing, security and environmental systems, communications and computer networks, elevators, etc. all working together. This is analogous to the dramatic increase in complexity that designers must tackle in electronics today.



Terminology

Silicon Intellectual Property:

Commonly known as *IP*. Reusable pre-made components for chips and circuits.

Models:

Stored designs that help in approximating the physical phenomenon of the circuit. Simple to moderately complex. Models are used by SPICE-type tools to help in design verification.

Foundry Libraries:

Or just *libraries*. Basic circuit building blocks, AND, OR, NOT, etc. Libraries are really an example of IP, but have much more simple functions.



Why add Encryption?

Protect Financial Investment:

Development of IP represents a significant financial investment. For example, a new foundry process can take millions of dollars to develop.

From:

- user error, e.g. due to inadvertent change or incorrect usage with an incompatible tool
- theft / competitor access

Example of User Errors:

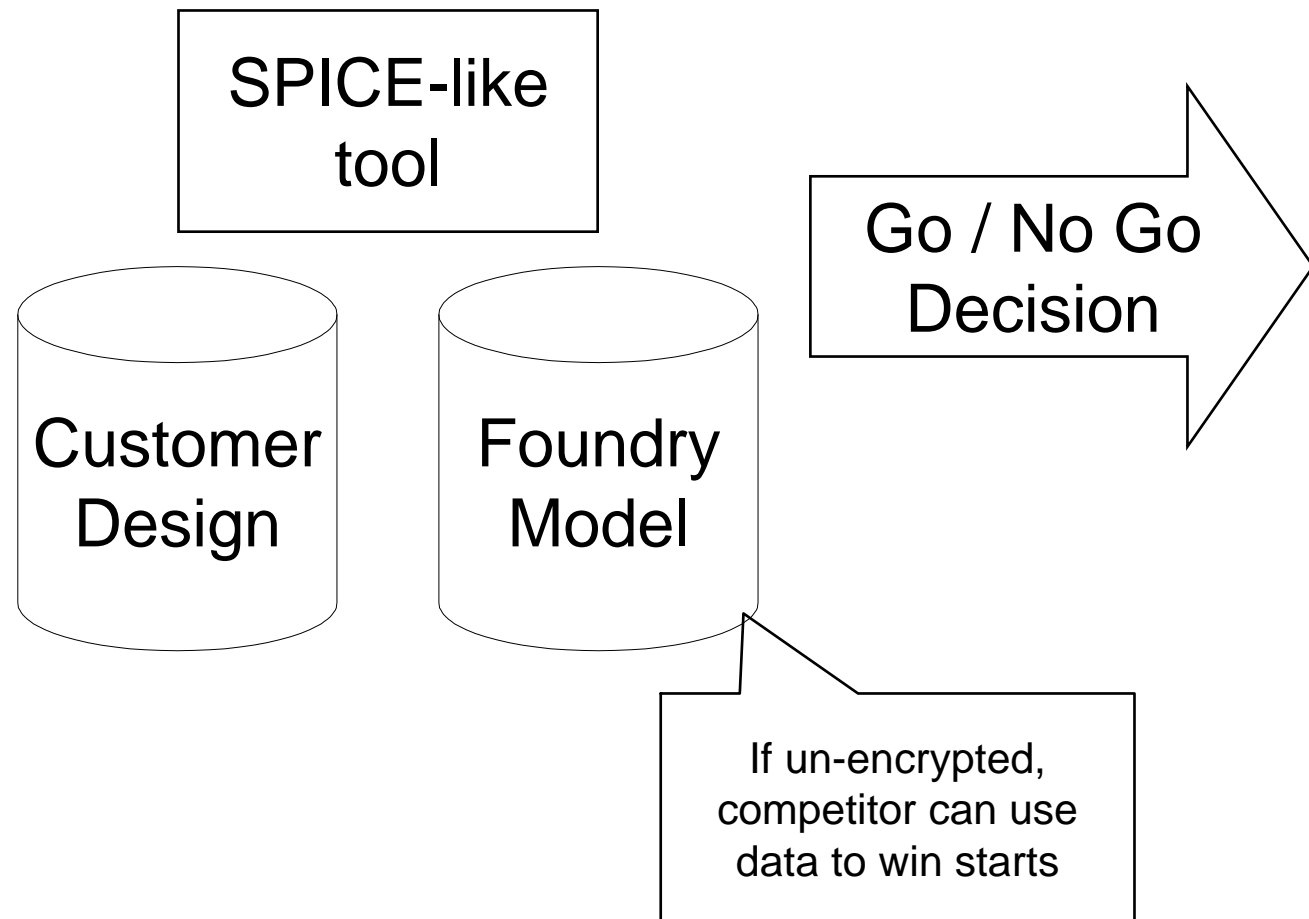
“Well [the user] didn’t like how that worked so [they] changed it...” (customer story related by EU-based IP provider)

Without encryption you can’t prevent the user from “breaking” the IP, that leads to support problems



- Foundries need to provide models to permit their customers to make go/no-go decisions on a design
- The model data provides a competitive advantage to the foundry, represents tens of millions of dollars of R&D effort
 - Need to protect “secret” sauce
 - Competitor access could allow free ride on R&D and allow them to steal design wins
- EDA tool providers implement encryption in the SPICE-like tool so that the foundry can encrypt the foundry model
- Customers can simulate the encrypted model, but cannot “see” it
- “Password” is typically embedded in the file*

Example 1: Model Encryption





Example 2: IP and Libraries

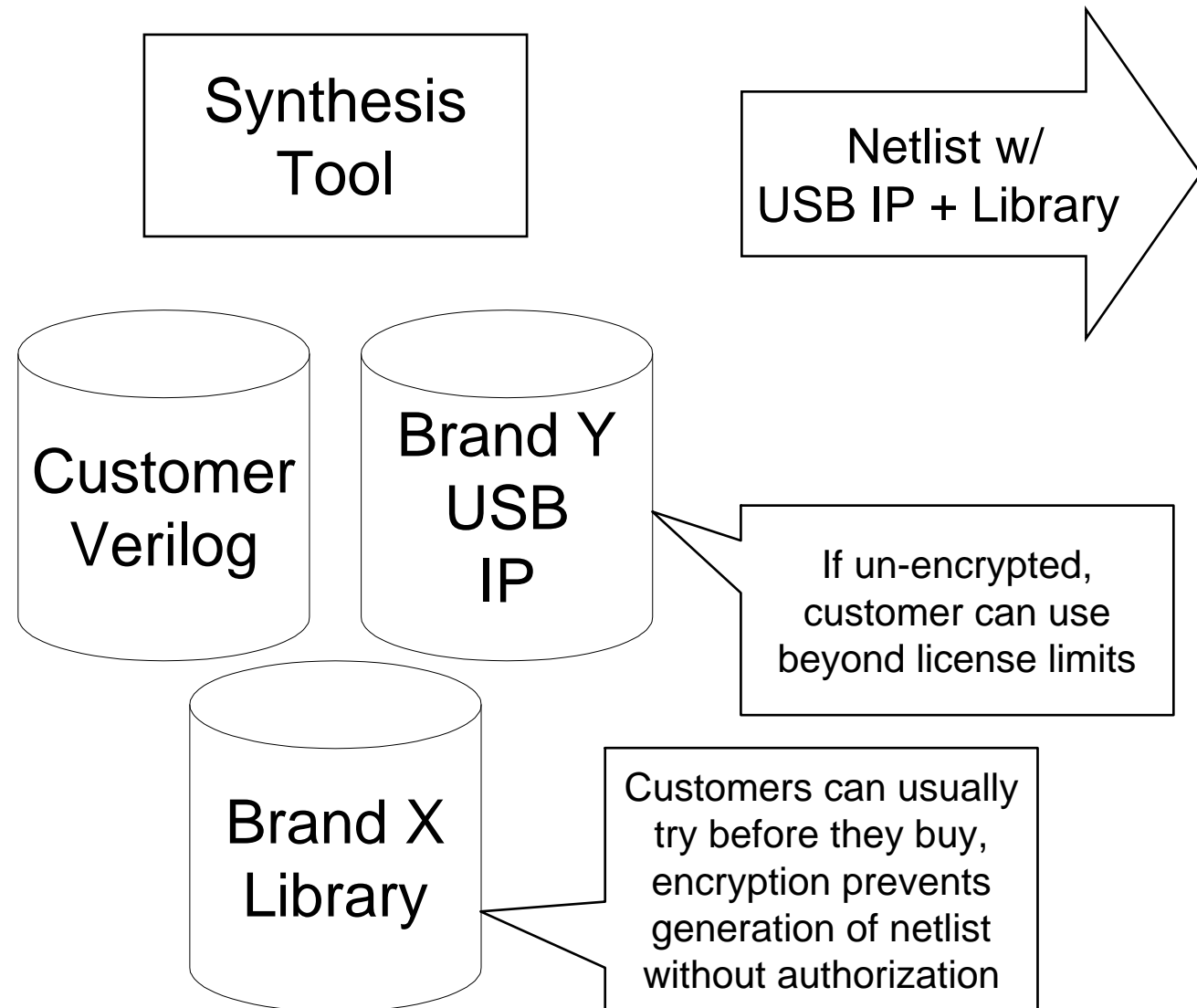
- Reuse of IP reduces design time, customers can focus their design efforts on the *functionality* they want their chip to implement, simply “invoke” the IP in their Verilog. Customers can use (simulate) the USB IP but not directly see the Verilog source code for it.

- Libraries may be differentiated by foundry/provider. Customers can try out the library but may not be able to generate a netlist for parts unless they have a full license from the foundry

- Encryption enforces license restriction, prevents theft of the IP

- Prevents misuse of IP, e.g. breaking the USB → might hurt seller’s reputation for reliability

- “Password” is typically in the file, but obscured*





Conclusion

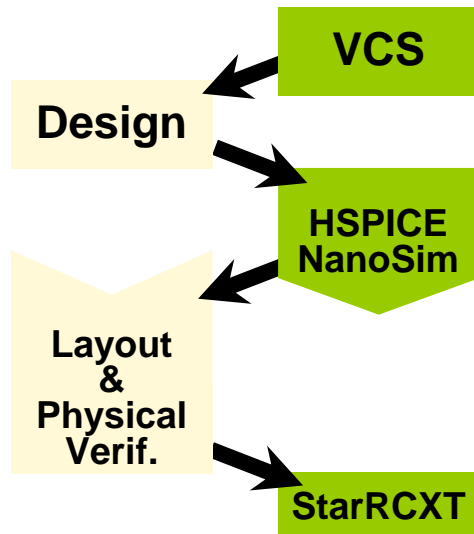
Presence of encryption for protection of customer designs and foundry information in EDA software is expected to expand to ever more EDA products due to customer demands

Examples of current encryption use follow



HSPICE Software (Synopsys, Inc.)

“Mixed Signal Flow”



HSPICE in a nutshell:

HSPICE is an analog circuit simulator. It understands the behavior of circuits, which process voltages and currents instead of 1's and 0's. At the lowest level, digital designs are composed of analog circuits. Analog designs can be integrated, or discrete. Integrated circuits are combined in chips or other packaging, while discrete designs are individual circuit elements (resistors, capacitors, inductors, transistors) combined on circuit boards and in systems.

Who needs HSPICE:

- IC designers who need to perform a sign-off prior to layout → **GO/NO GO for design to go to layout**
- Interconnect and signal integrity analysis engineers. To accurately analyze SI effects, HSPICE has the accurate models for the interconnections, drivers and receivers customers need.
- IC vendors use HSPICE to characterize new digital cell libraries and create timing and power models for downstream tools.
- Board-level designers of discrete analog circuits use HSPICE to verify functionality.

Product Information: <http://www.synopsys.com/products/mixedsignal/hspice/hspice.html>



Encryption in HSPICE

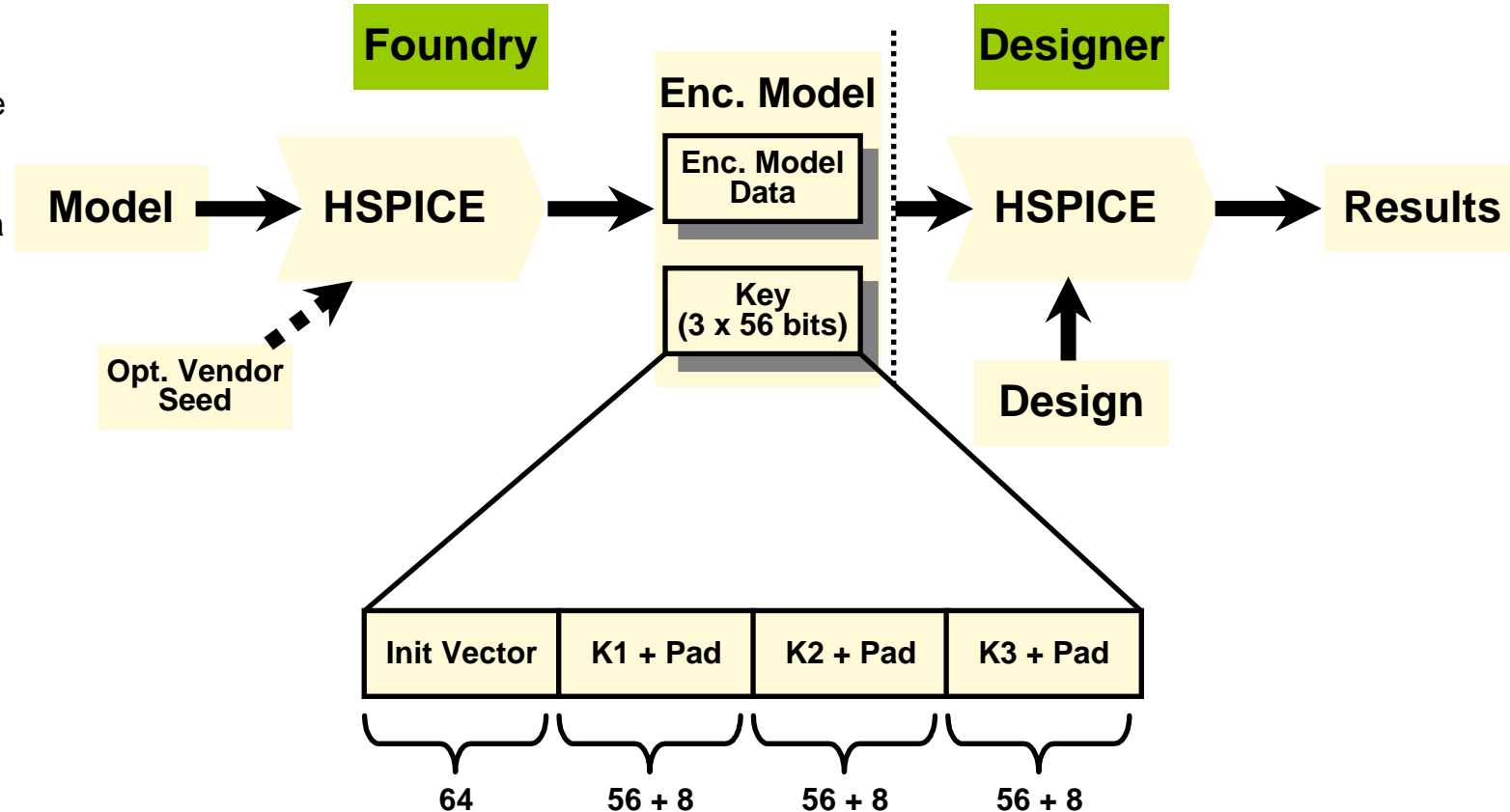
Two usage models:

- Key in file
- Optional out of band key exchange

In out of band key exchange mode, then the designer will be provided the encrypted model in two parts, the model data and the out of band key.

Optional vendor seed—or a random bit sequence—is permuted with a random seed to generate the Key.

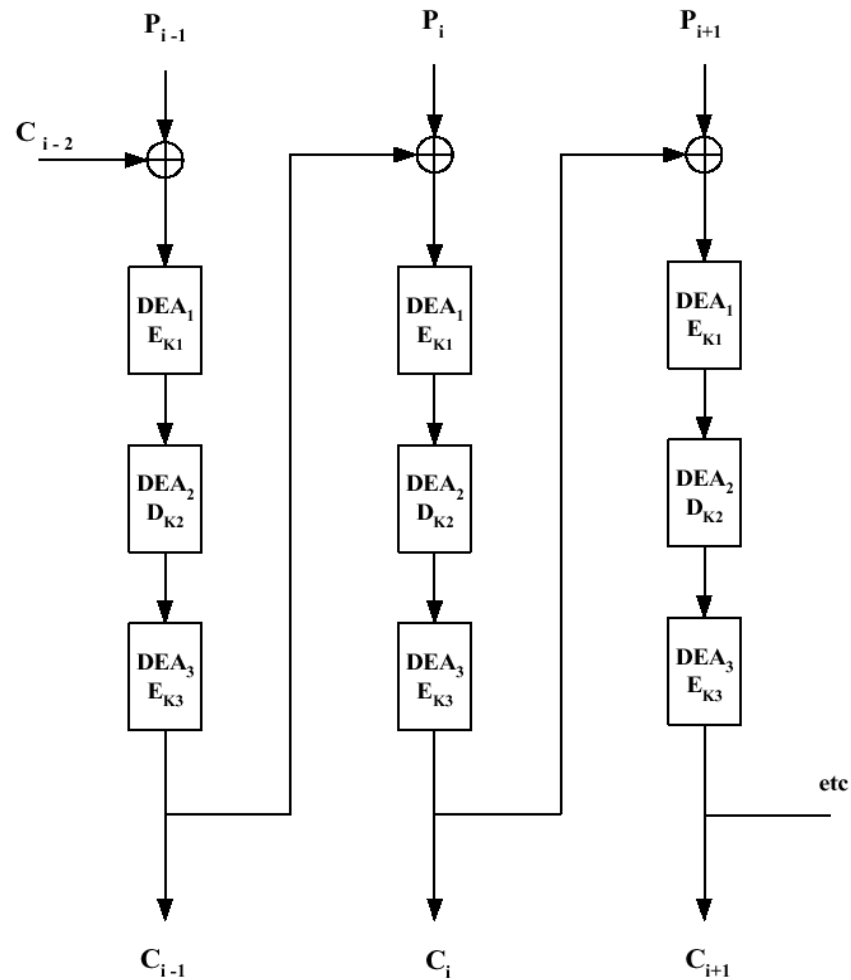
Fixed vendor seed will always produce different output even for the same model.





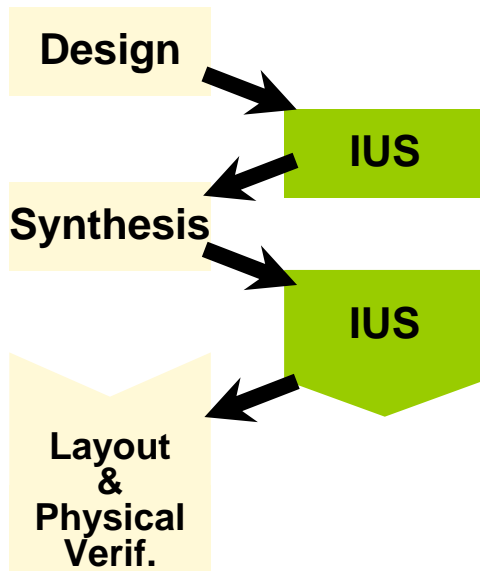
Textbook 3DES CBC Implementation

- Textbook 3DES CBC implementation: key includes initialization vector and three 56-bit keys, plus padding
- HSPICE can only encrypt/decrypt HSPICE data files, not arbitrary text/messages
- HSPICE in object code form only—no source; no user exposed cryptographic APIs
- Vendors expect us to ensure that HSPICE tool limits access of users, even with the key, to the underlying model data
- Out of band key mode solely a control on model use because “cracking” the key only provides use of the model; the “black hat” would still need to “crack” tool data structures to see “clear text”





“Verification Flow”



Incisive Unified Simulator (IUS) Software (Cadence Design Systems, Inc.)

Description of IUS:

Incisive Unified Simulator is a tool used to simulate digital circuits. The designs are represented using many different languages such as Verilog or VHDL. IUS supports those language as well additional languages used for specialized verification functions, such as SystemC, a derivative of C++. The tool handles any design that can be represented using a digital representation with the key languages. The Verilog only environment is called NC-Verilog and the VHDL one is called NC-VHDL. Designers depending on the complexity of their simulation tasks will create environments that use multiple languages to perform advanced verification tasks.

Who needs IUS:

- System architects who need to do analysis on various scenarios to determine what the right grouping of components would be. This is typically done with simple IP models to look at high level behavior.
- Design engineers who are creating the various parts of the circuit use IUS to test the behavior and make sure the requirements are met
- Verification Engineers are a specialized team that take the design once it is completed and create test that exercise the complete design testing actual conditions as best as possible.
- IP vendors used IUS to create IP models and ensure that their models behaves correctly with the tools that their customers will use.
- Board designers will use IUS as means to test the functionality of the board before it is built

Product Information: http://www.cadence.com/products/functional_ver/incisive_unified_simulator/index.aspx



NC-Protect is the encryption engine which is a standalone utility distributed with IUS

IUS can only run the encrypted model.

The tool is based on the premise that the user can never view the decrypted content

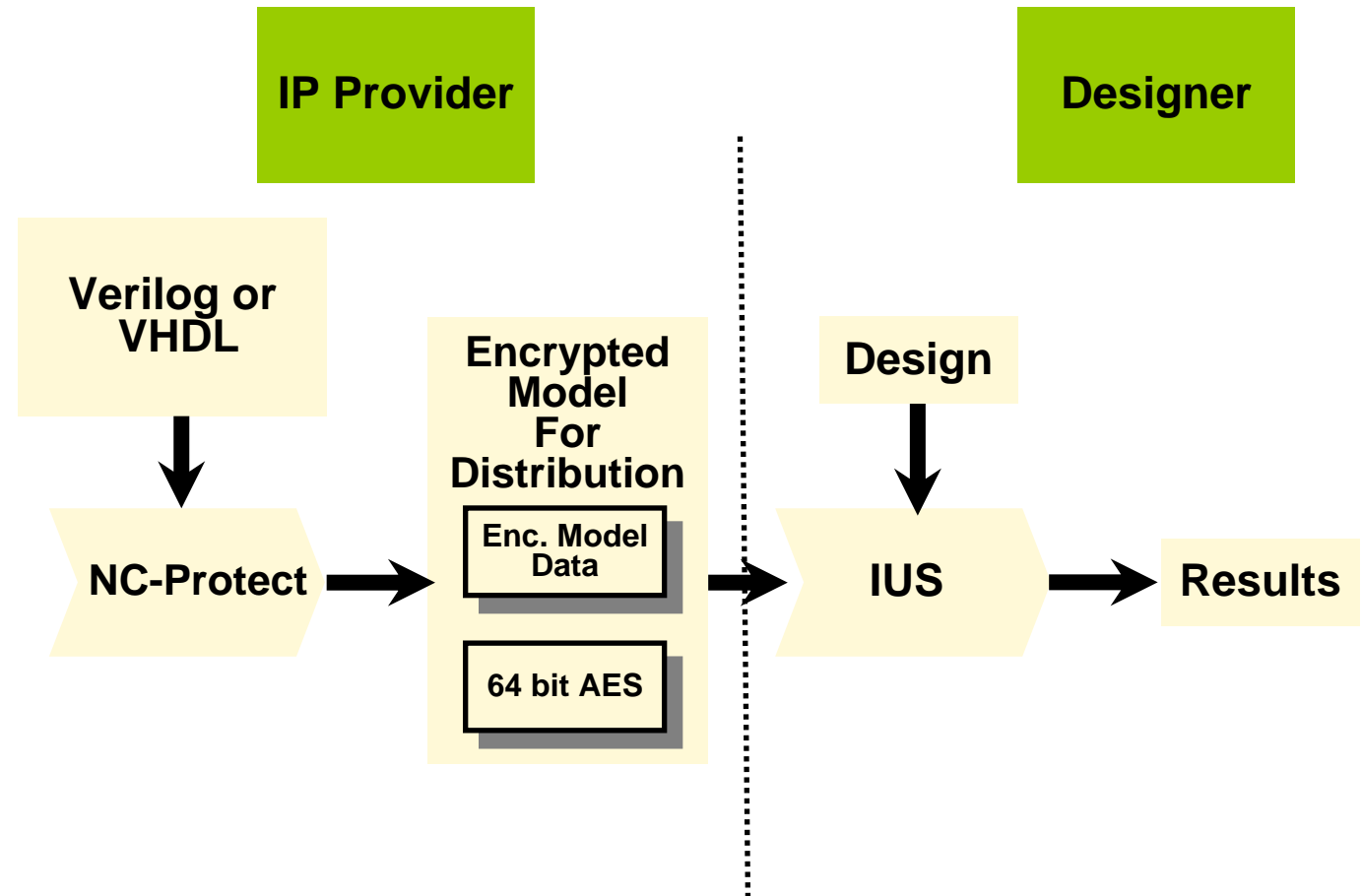
Only IUS supported languages can be run through NC-protect

Two usage models:

- Key in file
- Optional out of band key

If out of band key mode is used, then the designer will be provided the encrypted model in two parts, the model data and the key.

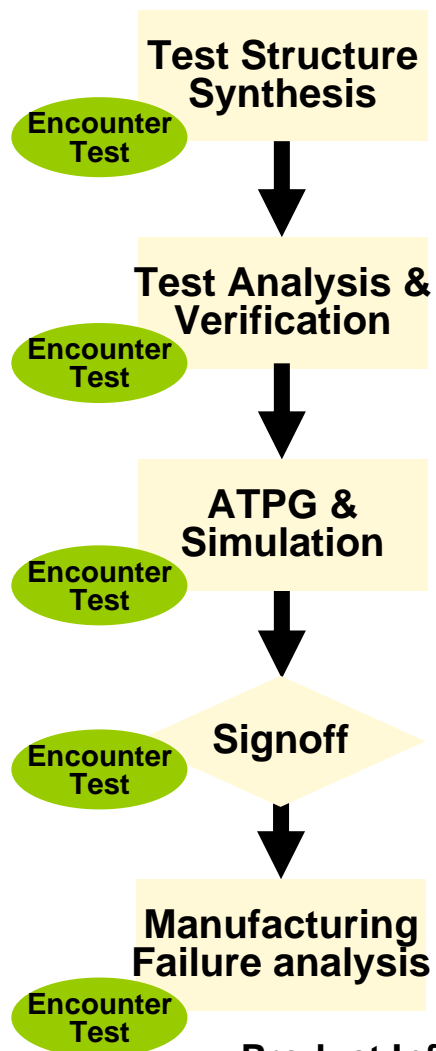
NC-Protect and IUS Software





Encounter Test & Encounter Test Model Protect Software (Cadence, Inc.)

“Encounter Test - Flow”



Encounter Test in a nutshell:

Encounter Test is a collection of tools that support the manufacture of chip designs in the following ways:

- Synthesize test logic structures into an existing chip design to enhance testability
- Produce a set of test patterns (or test vectors) to verify a chip was manufactured correctly
- Analyze failure data from chip manufacturing tests to determine where on the failed chip a defect might exist.

Who needs Encounter Test:

- Design engineers who are creating the various parts of the circuit use Encounter Test analysis tools to ensure their portion of the design can be tested.
- Test engineers are a specialized team that take the design once it is completed and use Encounter Test tools to generate patterns that test as many faults as possible on the manufactured chip.
- Chip fabricators use the Encounter Test generated test patterns (or test vectors) to stimulate and measure values on the manufactured chip.

When the chip fails (does not provide expected response values), Chip fabricators use Encounter Test tools to read in and analyze the test failures to identify areas of the chip where the defect causing the failure might exist.

Product Information: http://www.cadence.com/products/digital_ic/encountertest/index.aspx

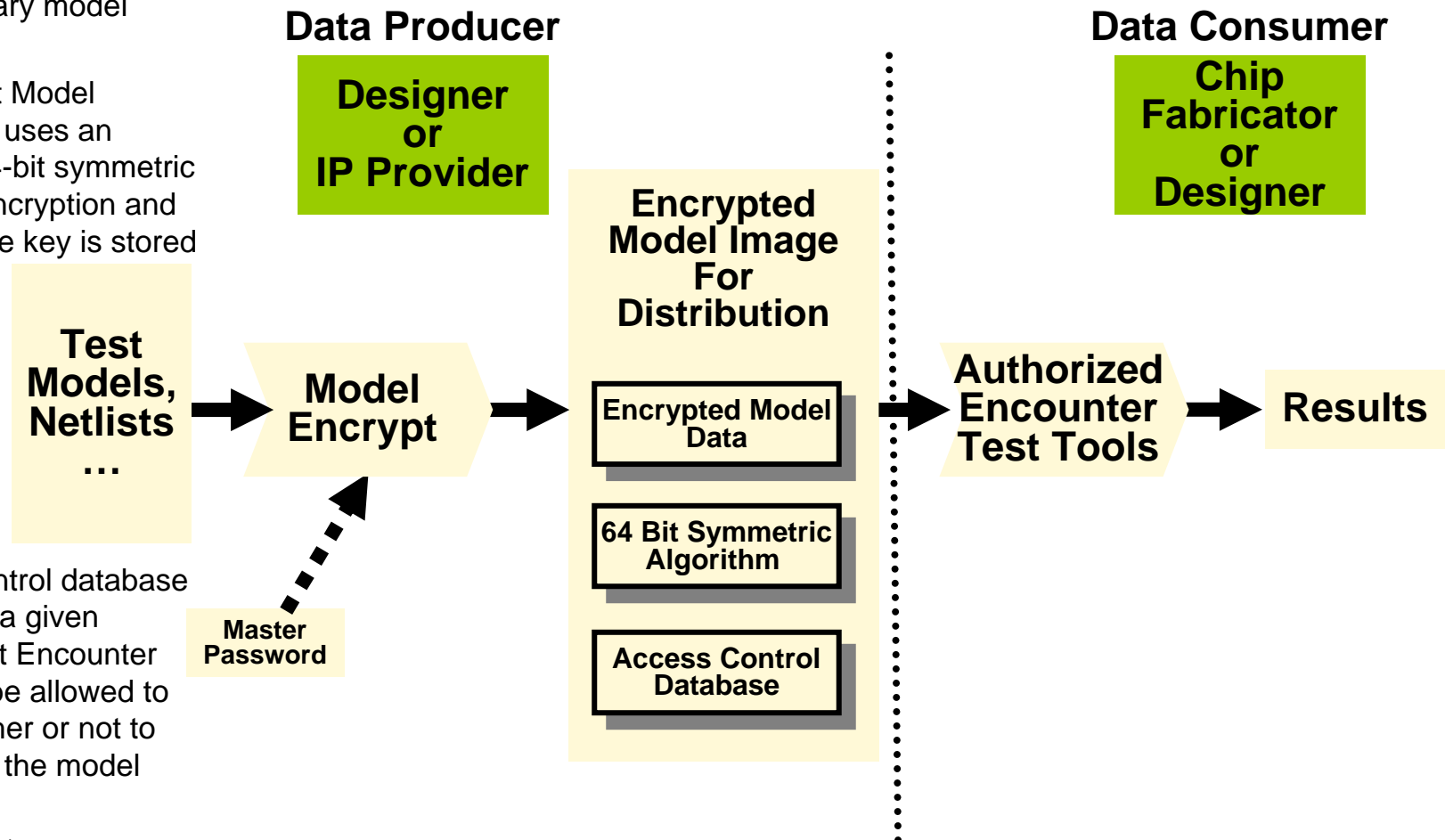


Encounter Test Model Protect Software

Encounter Test Model Protect, a separately licensed feature, encrypts meaningful text names in netlists and stores the result in a proprietary model database.

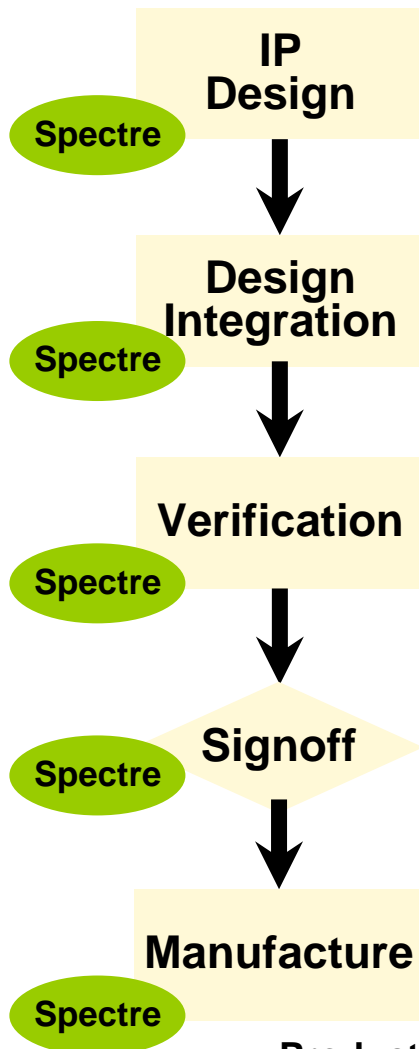
Encounter Test Model Protect feature uses an unpublished 64-bit symmetric algorithm for encryption and decryption. The key is stored with the model.

The access control database determines for a given password, what Encounter Test tools will be allowed to run, and, whether or not to decrypt data in the model





“A/MS and RF Flow”



Product Information: http://www.cadence.com/products/custom_ic/spectre/index.aspx

Spectre and Spectre Encrypt Software (Cadence, Inc.)

Spectre in a nutshell:

Virtuoso Spectre simulator provides fast, accurate transistor-level simulation for the Virtuoso custom design platform and provides detailed analysis in multiple domains (time, frequency, voltage, etc.)

These analyses provide early insight to actual function and are required as design engineers further refine design concepts into real designs that are headed for manufacturing.

In the diagram to the left, each “arrow” indicates that a company’s IP must be protected via encryption. The arrows can span companies and countries.

Who needs Spectre:

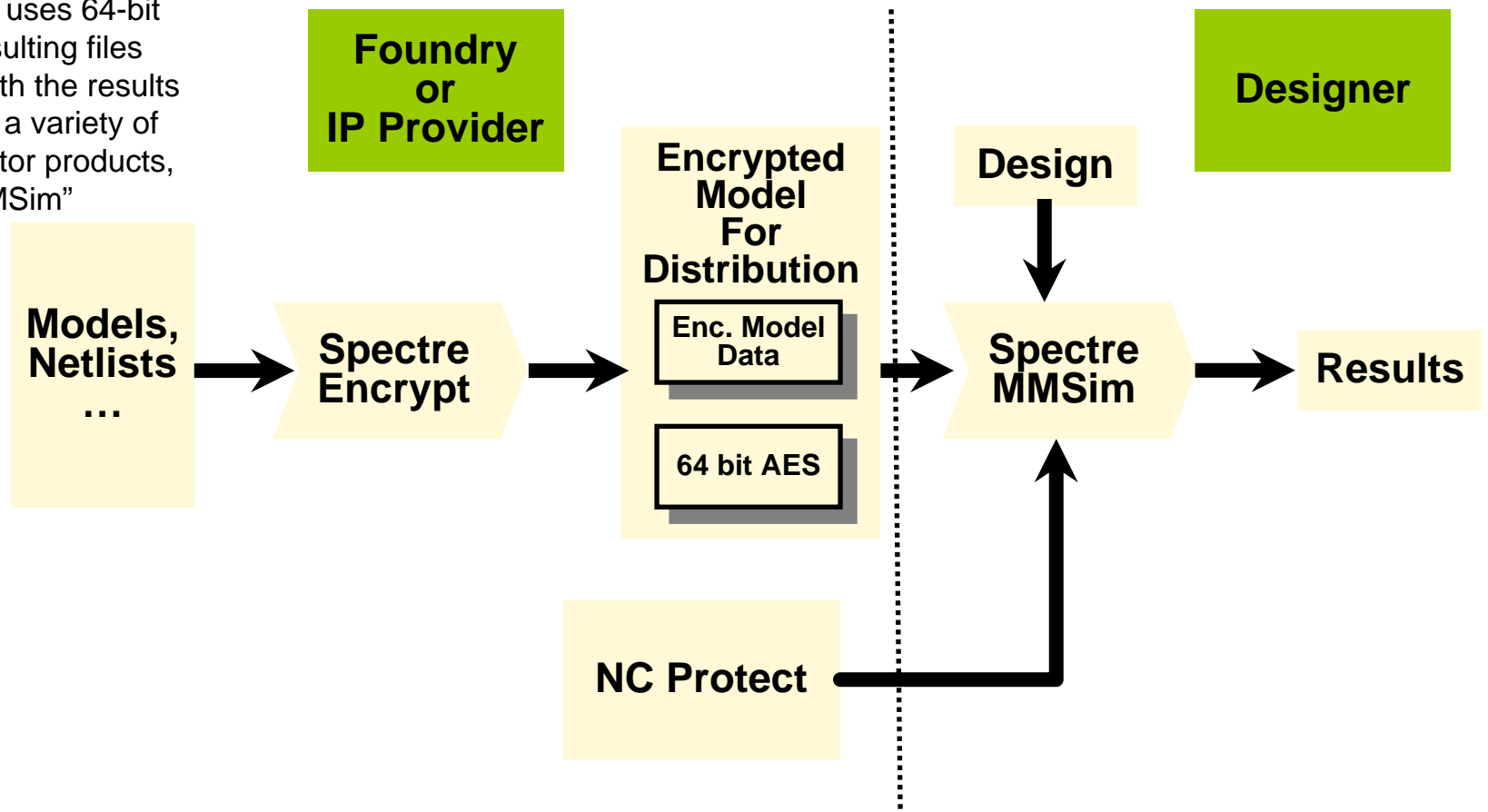
- Foundry process engineers creating new fabrication processes for semiconductor manufacturers.
- Design engineers working to create analog, mixed-signal and RF cells and blocks for larger semiconductor designs.
- Design engineers working towards signoff status for complex circuits.
- Design engineers working to include semiconductor components into larger systems.



Spectre Encrypt Software

Spectre Encrypt, a stand-alone, licensed product performs encryption of netlists, device model data or any other input format for Spectre.

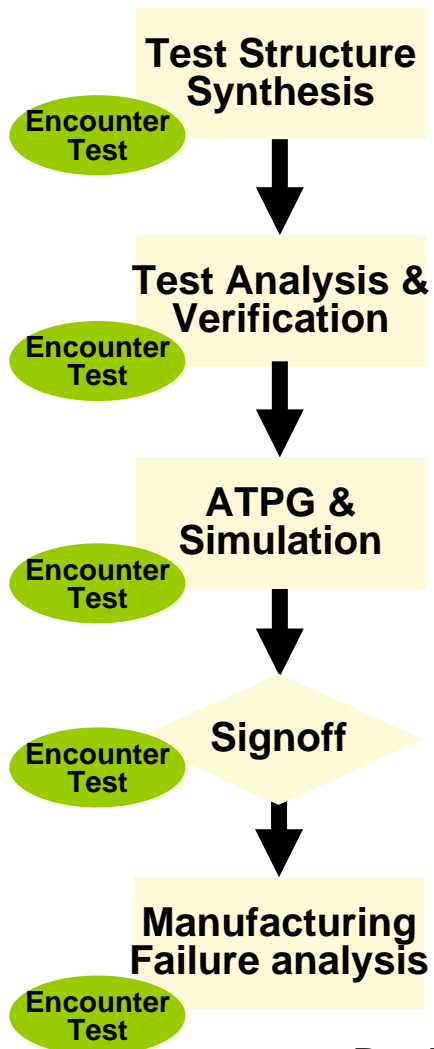
Spectre Encrypt uses 64-bit AES and the resulting files can be mixed with the results of NC Protect in a variety of Cadence simulator products, here called, "MMSim"





Encounter Test & Encounter Test Model Protect Software (Cadence, Inc.)

“Encounter Test - Flow”



Encounter Test in a nutshell:

Encounter Test is a collection of tools that support the manufacture of chip designs in the following ways:

- Synthesize test logic structures into an existing chip design to enhance testability
- Produce a set of test patterns (or test vectors) to verify a chip was manufactured correctly
- Analyze failure data from chip manufacturing tests to determine where on the failed chip a defect might exist.

Who needs Encounter Test:

- Design engineers who are creating the various parts of the circuit use Encounter Test analysis tools to ensure their portion of the design can be tested.
 - Test engineers are a specialized team that take the design once it is completed and use Encounter Test tools to generate patterns that test as many faults as possible on the manufactured chip.
 - Chip fabricators use the Encounter Test generated test patterns (or test vectors) to stimulate and measure values on the manufactured chip.
- When the chip fails (does not provide expected response values), Chip fabricators use Encounter Test tools to read in and analyze the test failures to identify areas of the chip where the defect causing the failure might exist.

Product Information: http://www.cadence.com/products/digital_ic/index.aspx?lid=dic

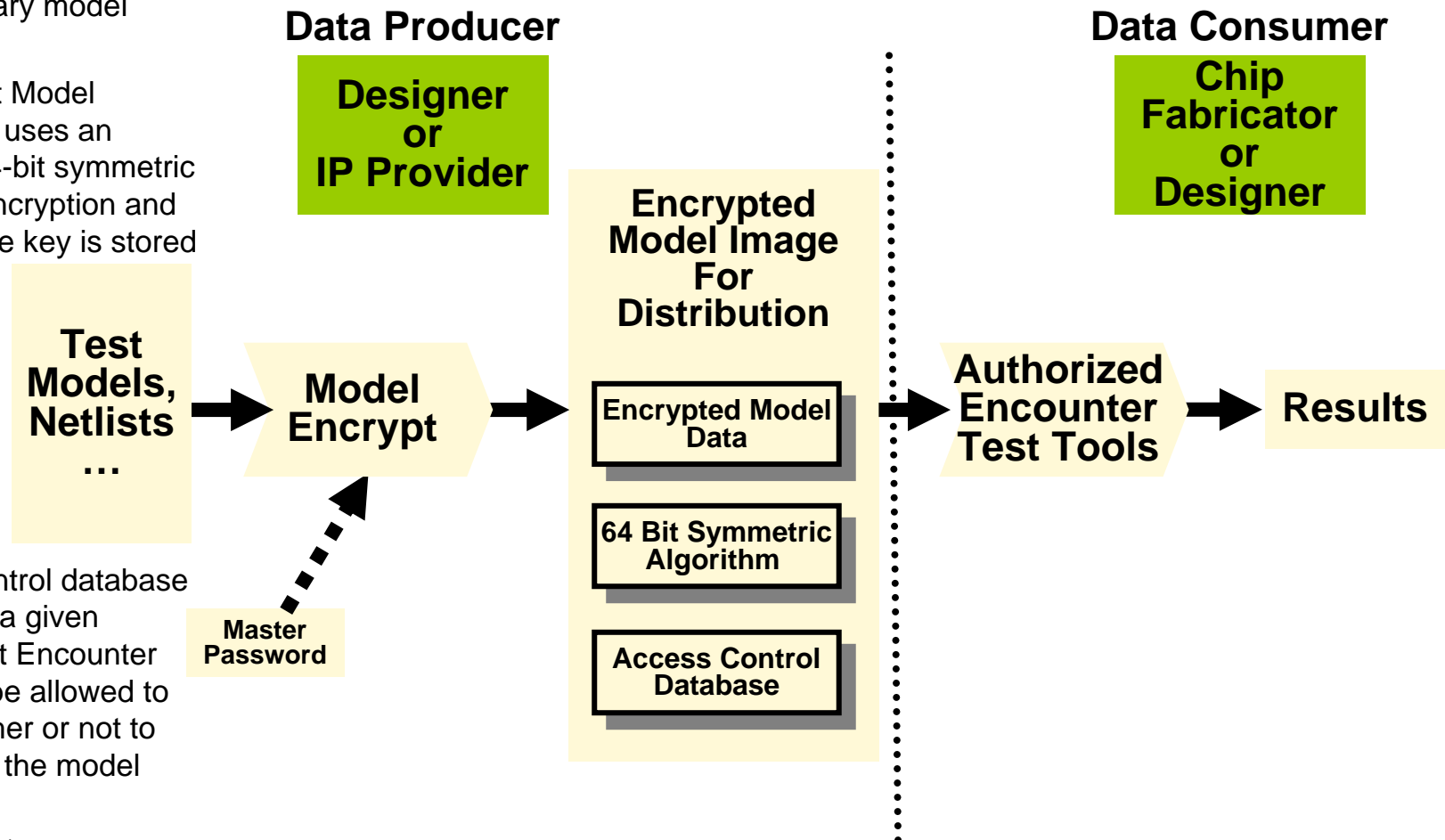


Encounter Test Model Protect Software

Encounter Test Model Protect, a separately licensed feature, encrypts meaningful text names in netlists and stores the result in a proprietary model database.

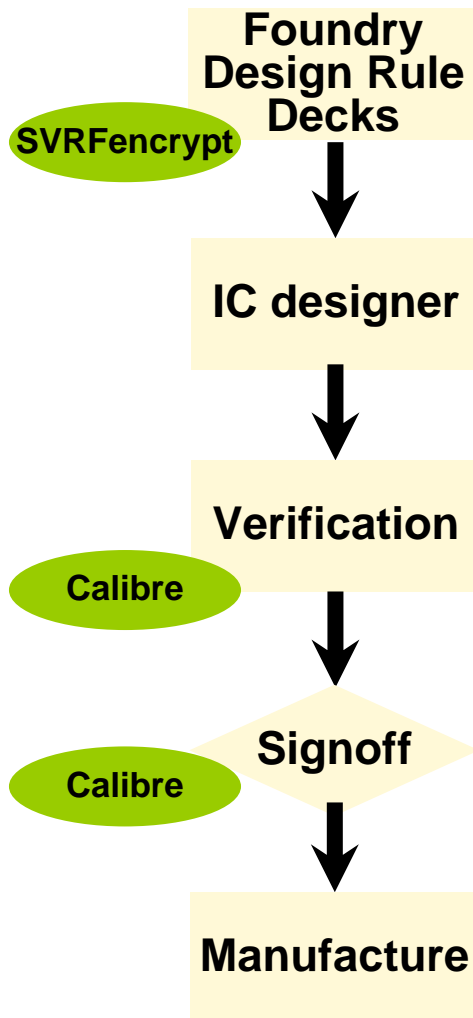
Encounter Test Model Protect feature uses an unpublished 64-bit symmetric algorithm for encryption and decryption. The key is stored with the model.

The access control database determines for a given password, what Encounter Test tools will be allowed to run, and, whether or not to decrypt data in the model





“Calibre Flow”



Calibre and Calibre SVRFencrypt Software (Mentor Graphics Corp.)

Calibre in a nutshell:

The Calibre product line is used for deep submicron physical verification and sub-wavelength manufacturability. It offers fast and reliable solutions to design rule checking (DRC), layout vs. schematic (LVS), silicon vs. layout, and electrical rule checking (ERC).

These verification tools compare the design to technical information from the IC foundry (design rule decks) to predict whether the design concepts will work when the completed designs are transferred to manufacturing.

Who needs Calibre:

- Foundry process engineers creating new fabrication processes for semiconductor manufacturers.
- Design engineers working to create larger semiconductor designs.
- Design engineers working towards signoff status for complex circuits.



The IC foundry encrypts their rule decks using Calibre SVRFencrypt, and provides the encrypted deck to the end-user. The encrypted rule deck includes the decryption key, in encrypted form.

The IC designer inputs their design and the encrypted rule deck into a Calibre product. Calibre has embedded in it a decryption key to decrypt the key in the rule deck. Calibre decrypts the key and the rule deck, performs the verification, and outputs a verification result; the user never has access to the clear text rule deck, or to any of the encryption functions.

Calibre uses 256-bit AES encryption.

Encryption in Calibre Software

