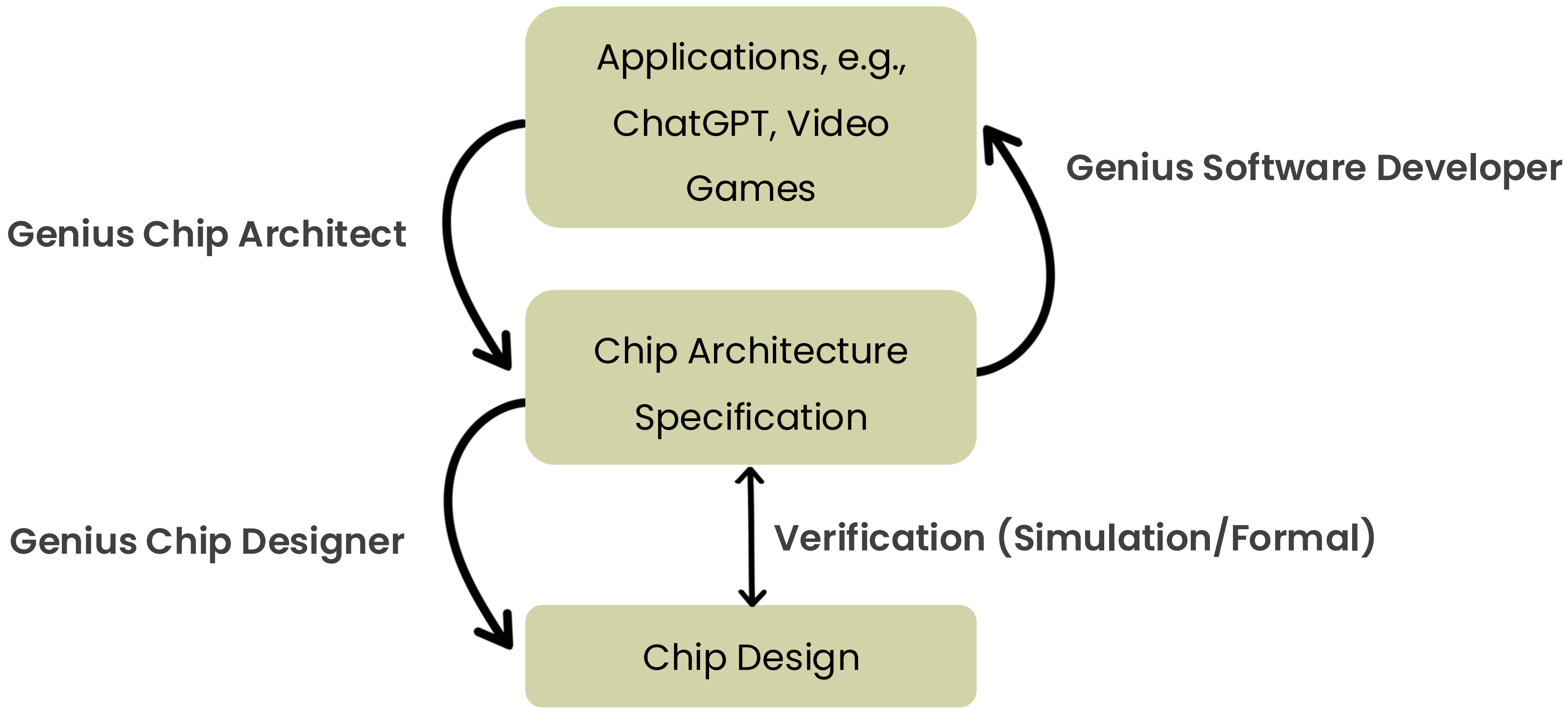
LUBIS EDA

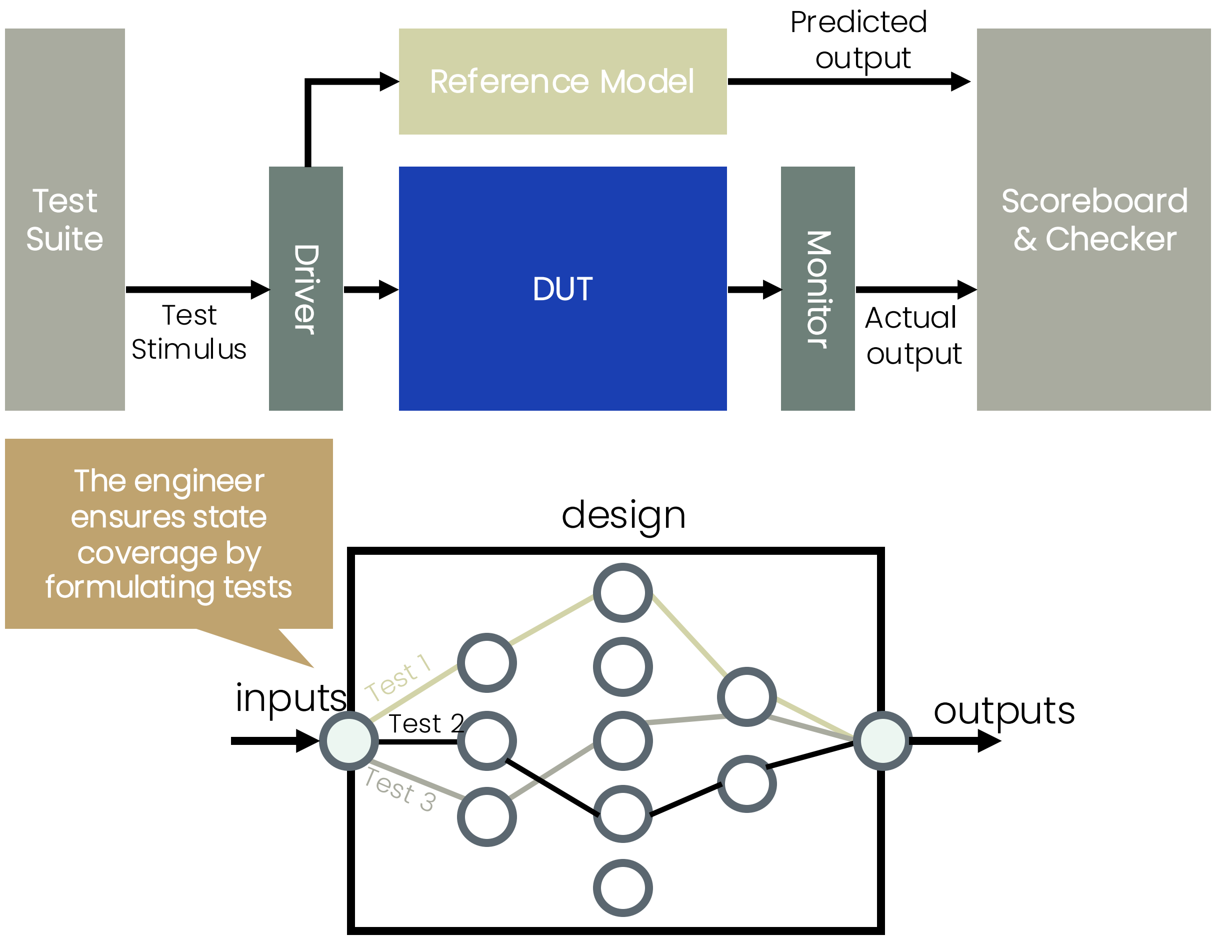
From Abstract Model to RTL: Automated Formal Assertions for Correct-by-Construction Design

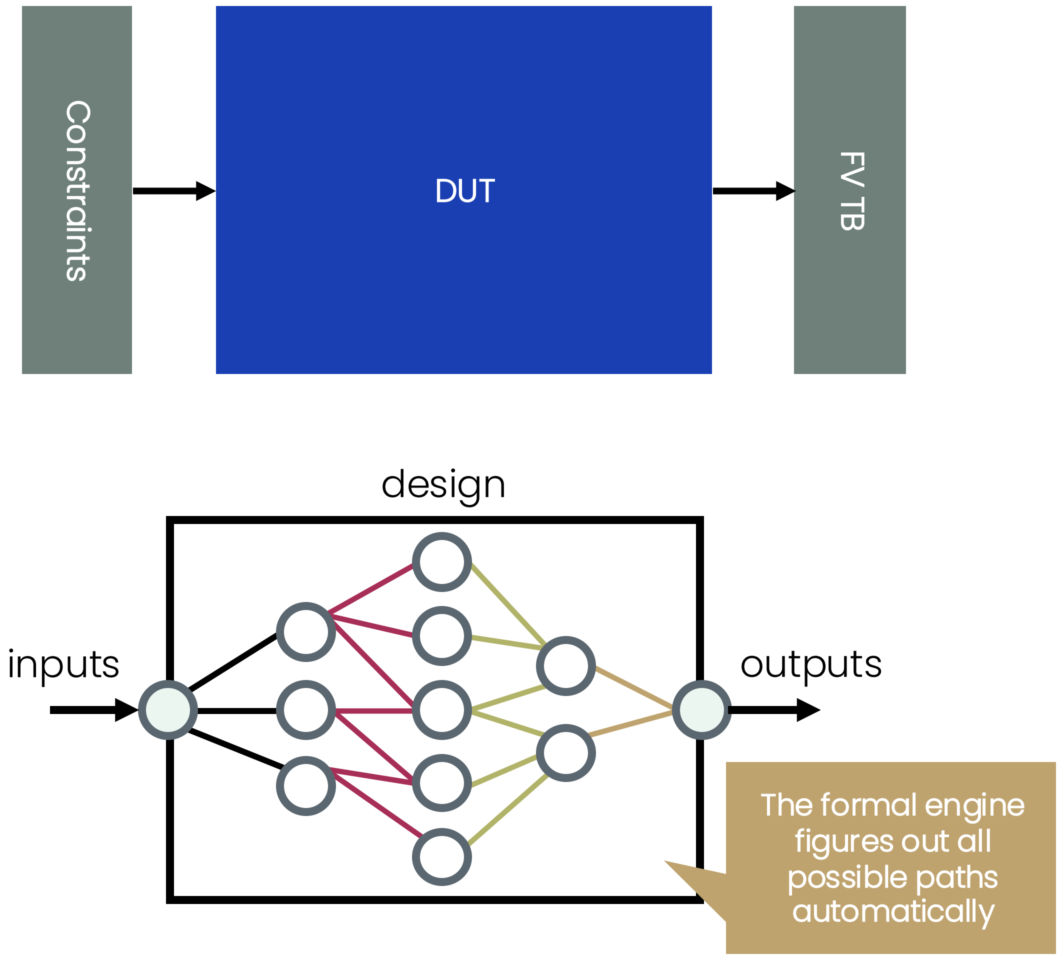
What it takes to have a successful chip?

Tens of thousands of engineer days



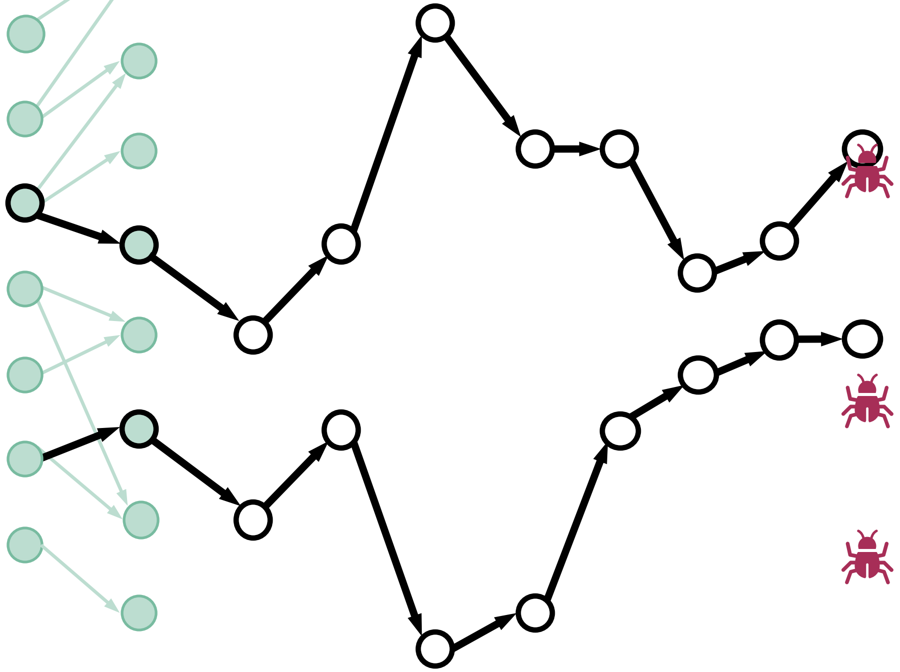
Simulation VS Formal



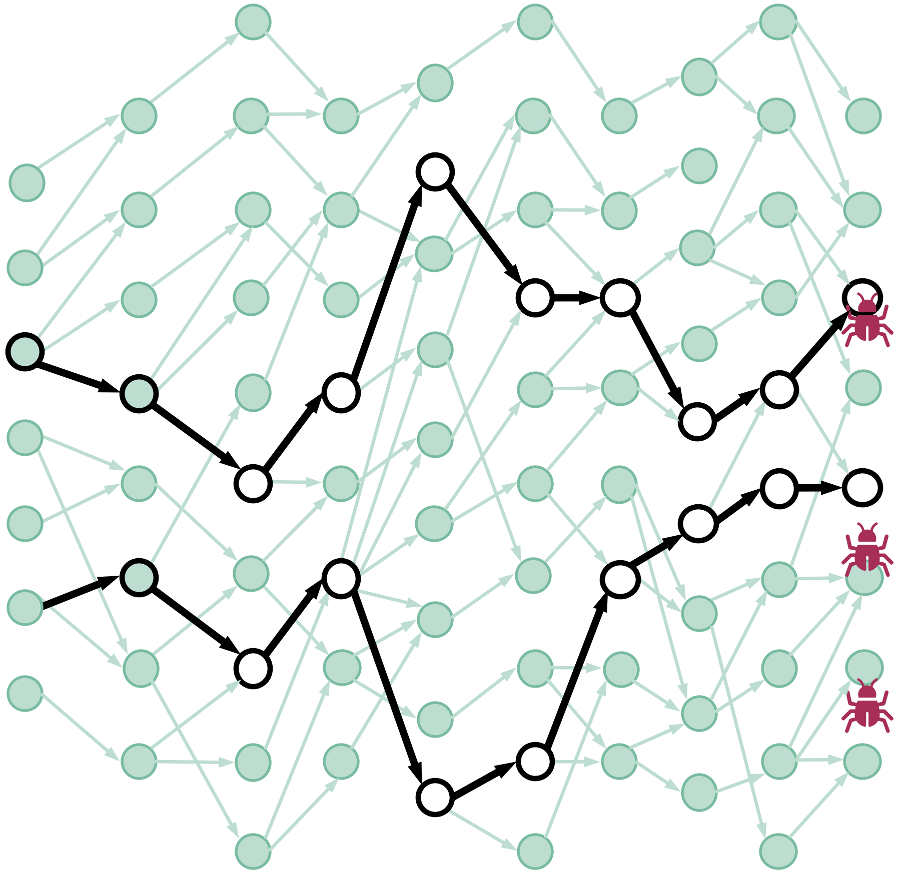


Simulation VS Formal

UVM

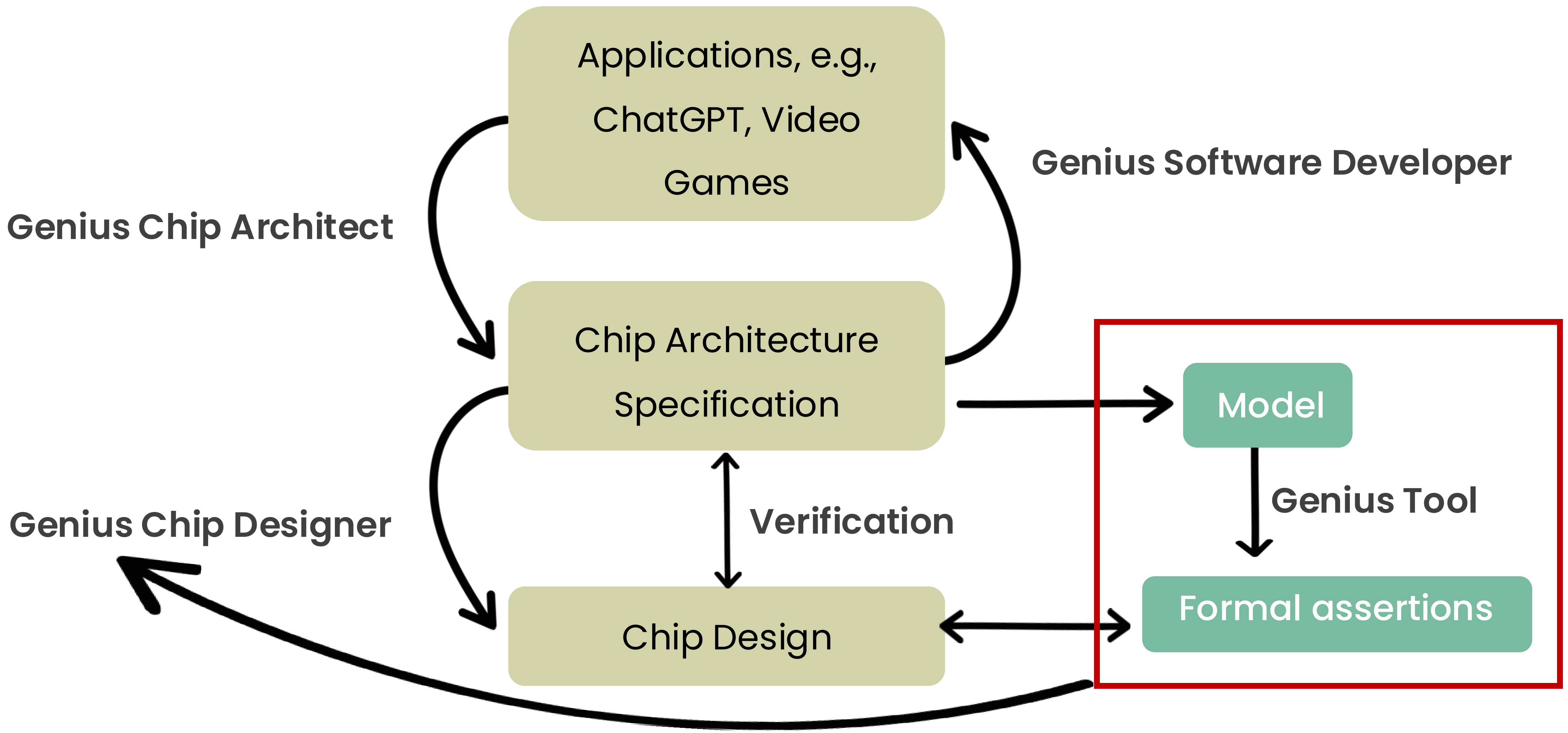


Formal



What it takes to have a successful chip?

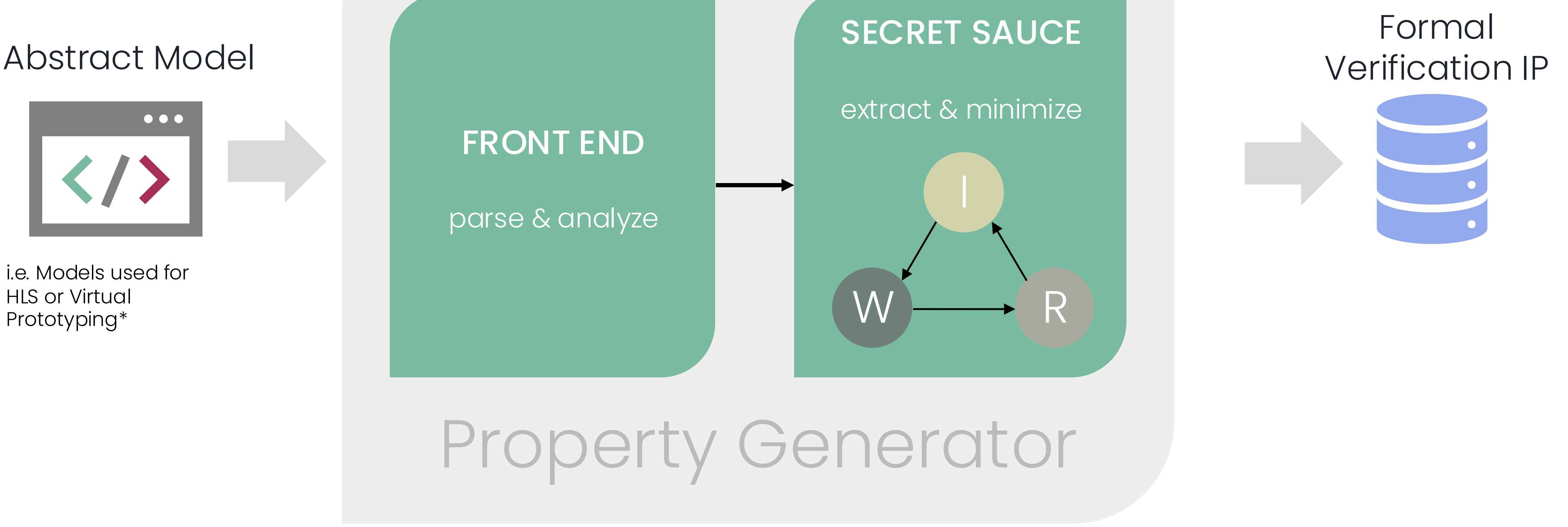
Tens of thousands of engineer days



Guard railing the chip designer.

Property Generator

From model to verification IP in minutes



* HLS and Virtual Prototyping models might not work out of the box and require some changes

Why automate Property Generation

From abstract models to guaranteed sign-off — automatically

Abstract Models as a Source

Complete and Correct by Construction

Reuse of Existing Models*

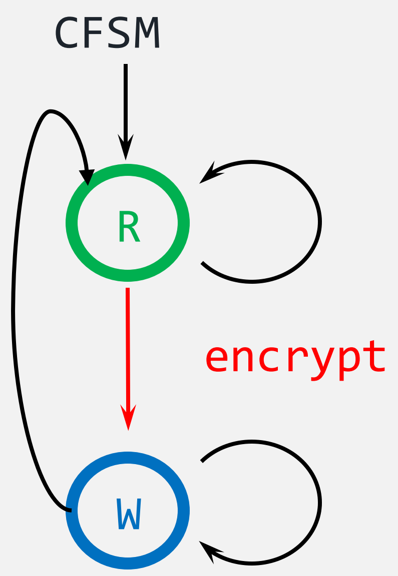
Simple Proof Complexity

*Existing models might not work out of the box and could require some changes

Modelling

Conceptualize the abstract behavior of the design as a model

```
SC_MODULE(encrypt) {
    // constructor and declarations
   void fsm() {
      while(true){
       input→read(data);
       output→write(encrypt(data));
       // some code
    } } };
```

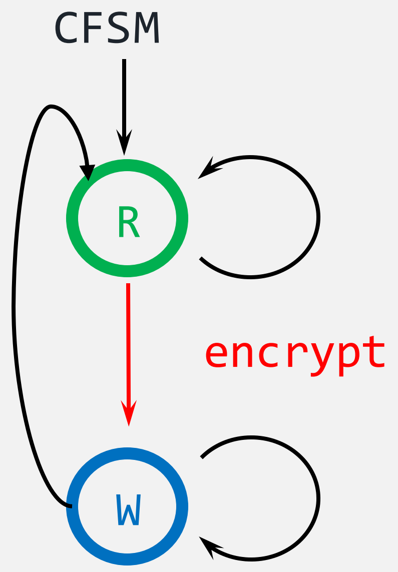


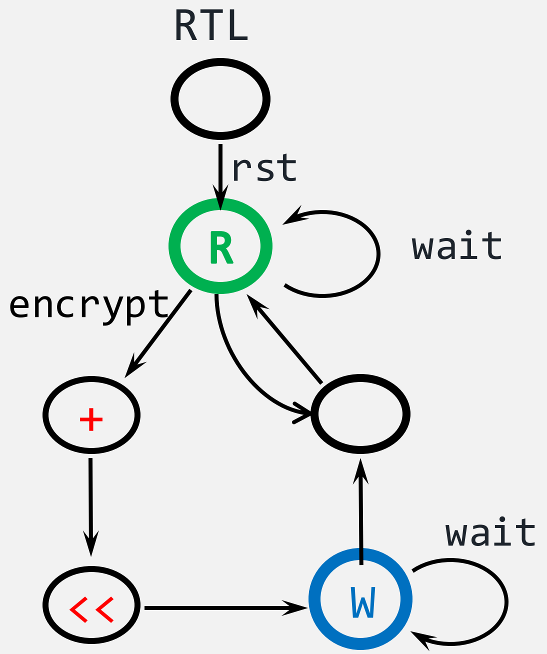

```
Property encrypt;
assume:
  at t: state == R;
  at t: data_valid == 1;
prove:
  at t+n: out(enc(data@t));
  at t+n: state == W;
endproperty;
```

Abstraction to Implementation

Match the model with the concrete implementation

```
SC_MODULE(encrypt) {
    // constructor and declarations
    void fsm() {
      while(true){
       input→read(data);
       output→write(encrypt(data));
       // some code
    } } };
```





Model to Property

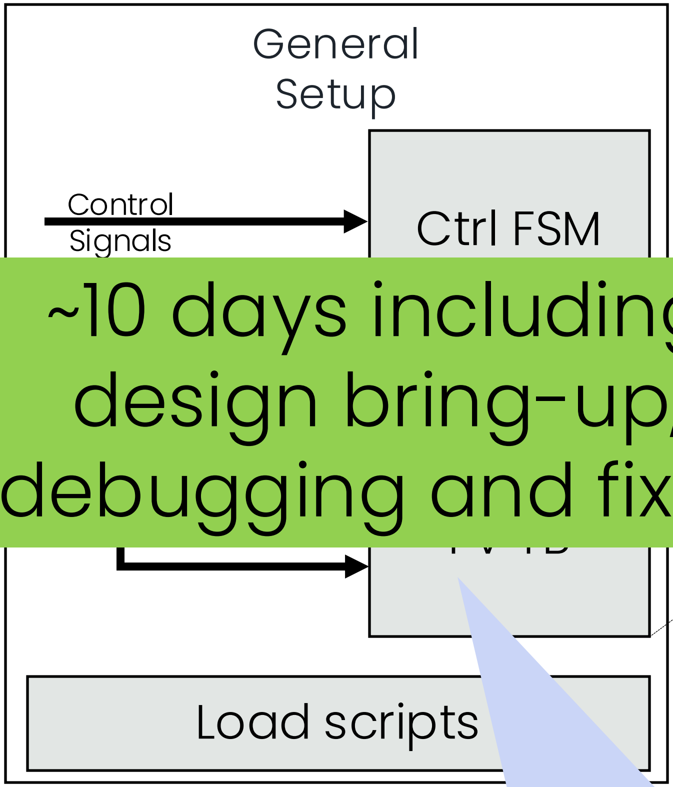
```
SHA_Input->read(SHA_in, “IDLE”);

if(init){
	// Some Computation
}
// Some Computations
for(i = 0; i < NUM_ROUNDS; ++i){
  insert_state(“SHA_Rounds”);
  if(i < 16){
    // Some Computation
  } else {
    // Some Computation
  }
  // Some Computation
}
insert_state(“DONE”);
```

```
property fsm_SHA_Rounds_to_SHA_Rounds_1_p;
	SHA_Rounds &&
	i >= 16 &&
	((i + 1) < 80)
|->
	##1 ((SHA_Input_rdy == 0) &&
		(out_vld == 0)) and
	##1
	SHA_Rounds &&
	$stable(H) &&
	W == $past(W_2_i, 1) &&
	a == ($past(T1_1_i, 1) + $past(T2_0_i, 1)) &&
	b == $past(a, 1) &&
	c == $past(b, 1) &&
	d == $past(c, 1) &&
	e == ($past(d, 1) + $past(T1_1_i, 1)) &&
	f == $past(e, 1) &&
	g == $past(f, 1) &&
	h == $past(g, 1) &&
	i == (7'd1 + $past(i, 1));
endproperty
```

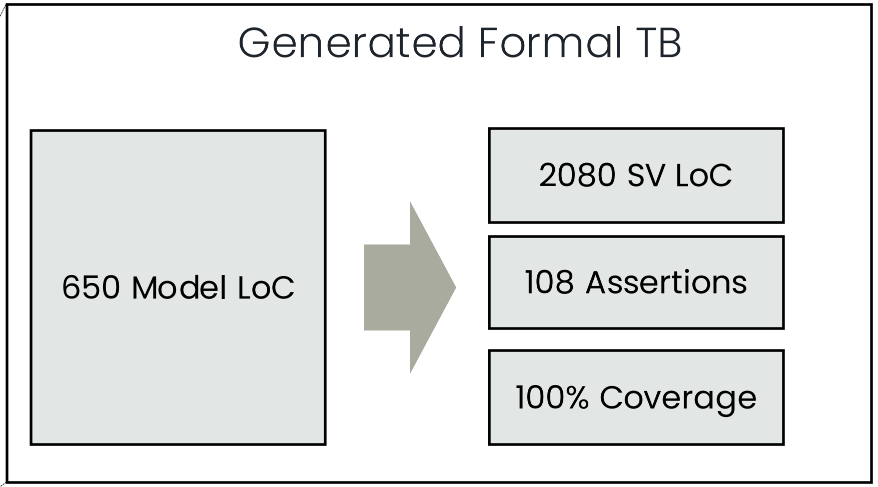
Success Story

Control FSM with 30+ states - TOP3 US SEMI



~10 days including design bring-up, debugging and fixes

Generated with LUBIS VIP Generation

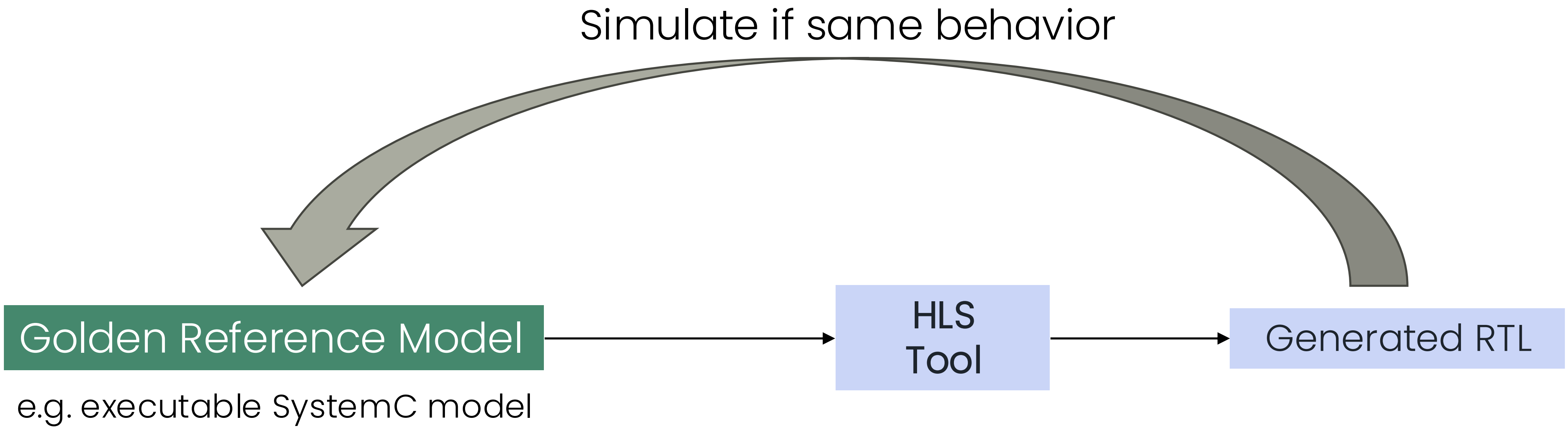


35+ Bugs found (~8 critical and corner case bugs)

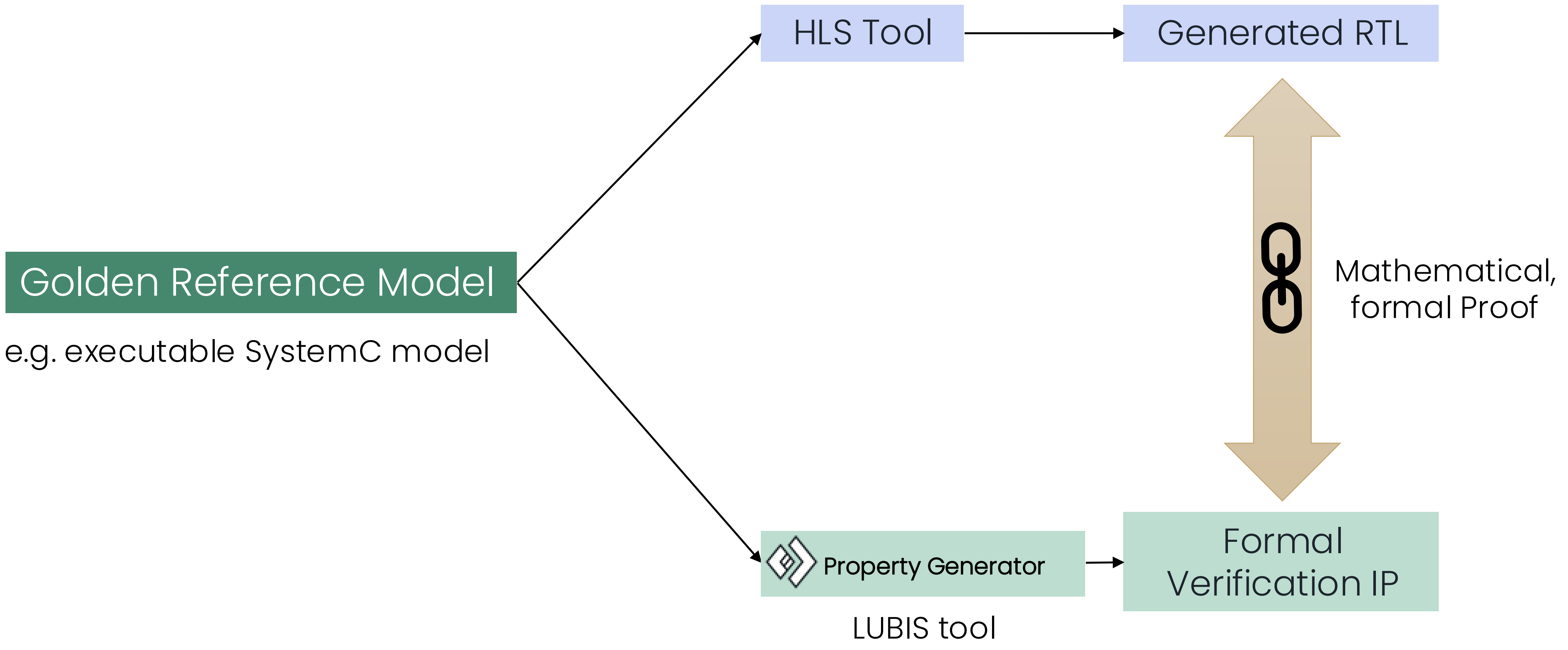
Detected bugs:
Several bugs were detected in the control logic, deadlocks, livelocks, deadcode, etc.

High-Level-Synthesis (HLS)

Status Quo



High-Level-Synthesis (HLS) with Formal Verification



We help chip teams reach formal sign-off

1 **Transforming** FV from an art into a system

2 **Services & Training** to achieve Formal Sign-Off

3 **Technology R&D** to enhance productivity in Formal

Team
35+

Kaiserslautern,
Germany

We love formal

since 2020

Want to learn more?

Follow us to get notified on news about Formal Verification solutions

Osama Ayoub

Technical Sales and Formal Verification Engineer

Email: osama.ayoub@lubis-eda.com

Web: www.lubis-eda.com

LinkedIn