



Under the Hood

Günter Bartsch

zamiaCAD Architecture

Eclipse Plugin

Analysis+Navigation

- Signal tracing
- Reference tracing
- Outlines, trees
- Path handling

Waveform + Simulation

- VCD import
- Built-in simulator
- Annotation

Design Entry

- Incremental builder
- Completion
- Error markers

FS Cache

Indexer,
Parser

Syntax Tree

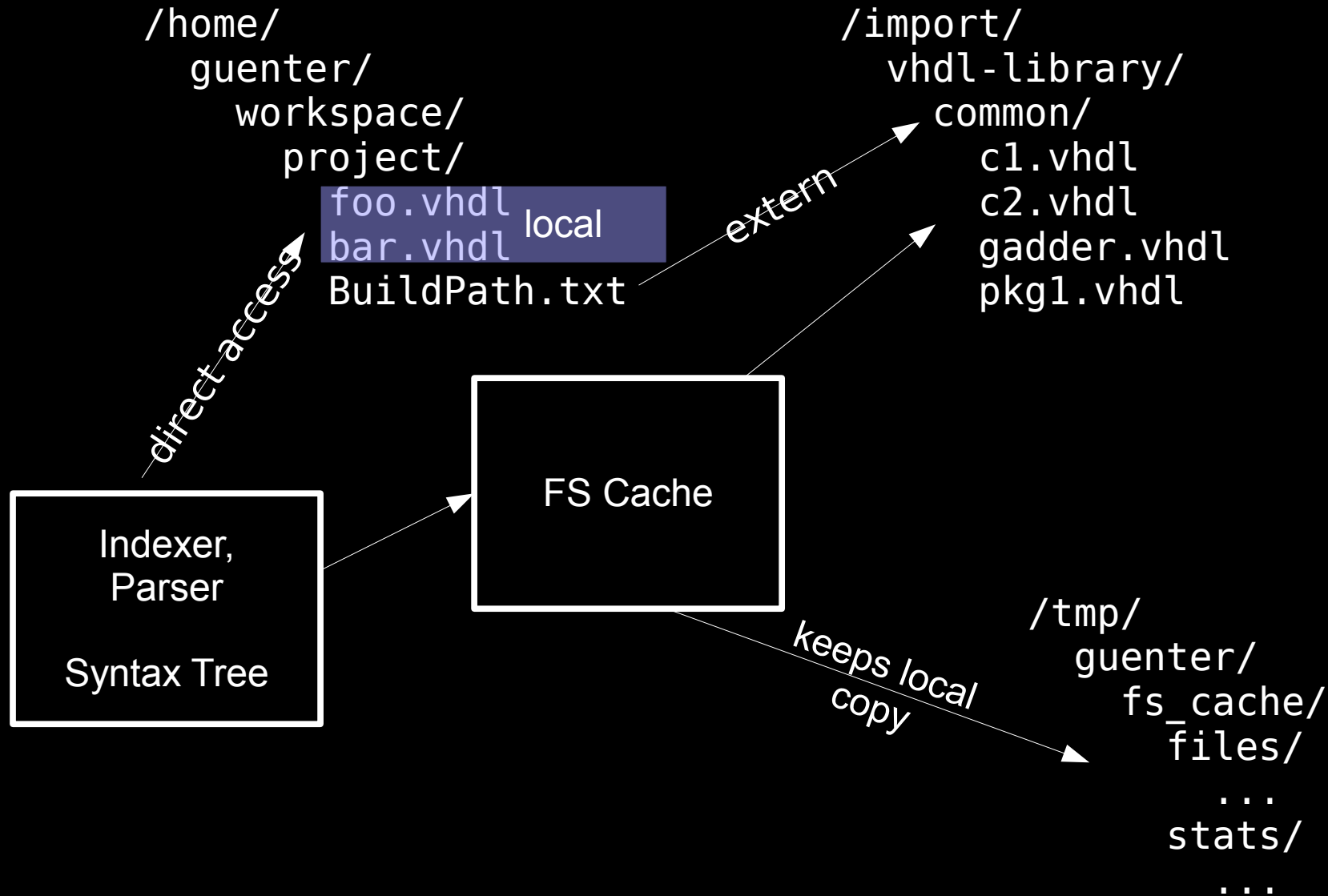
Elaboration
Engine

Model



ZDB

FSCache



VHDL Indexer

- pure lexer, (nearly) as fast as grep
- Multi-Threaded (helps with AFS latencies)

ahbctrl.vhd

```
use std.textio.all;
-- pragma translate_on

entity ahbctrl is
  generic (
    defmast      : integer := 0;
    ...
  );
end;

architecture rtl of ahbctrl is
  constant nahbm : integer := 2**log2(nahbm);
  ...
```

VHDL
Indexer

```
ahbctrl.vhd <-> ENTITY_AHBCTRL
ahbctrl.vhd <-> ARCH_AHBCTRL

ENTITY_AHBCTRL implemented by
ARCH_AHBCTRL
```

Demo

zamiaCAD Architecture

Eclipse Plugin

Analysis+Navigation

- Signal tracing
- Reference tracing
- Outlines, trees
- Path handling

Waveform + Simulation

- VCD import
- Built-in simulator
- Annotation

Design Entry

- Incremental builder
- Completion
- Error markers

FS Cache

Indexer,
Parser

Syntax Tree

Elaboration
Engine

Model

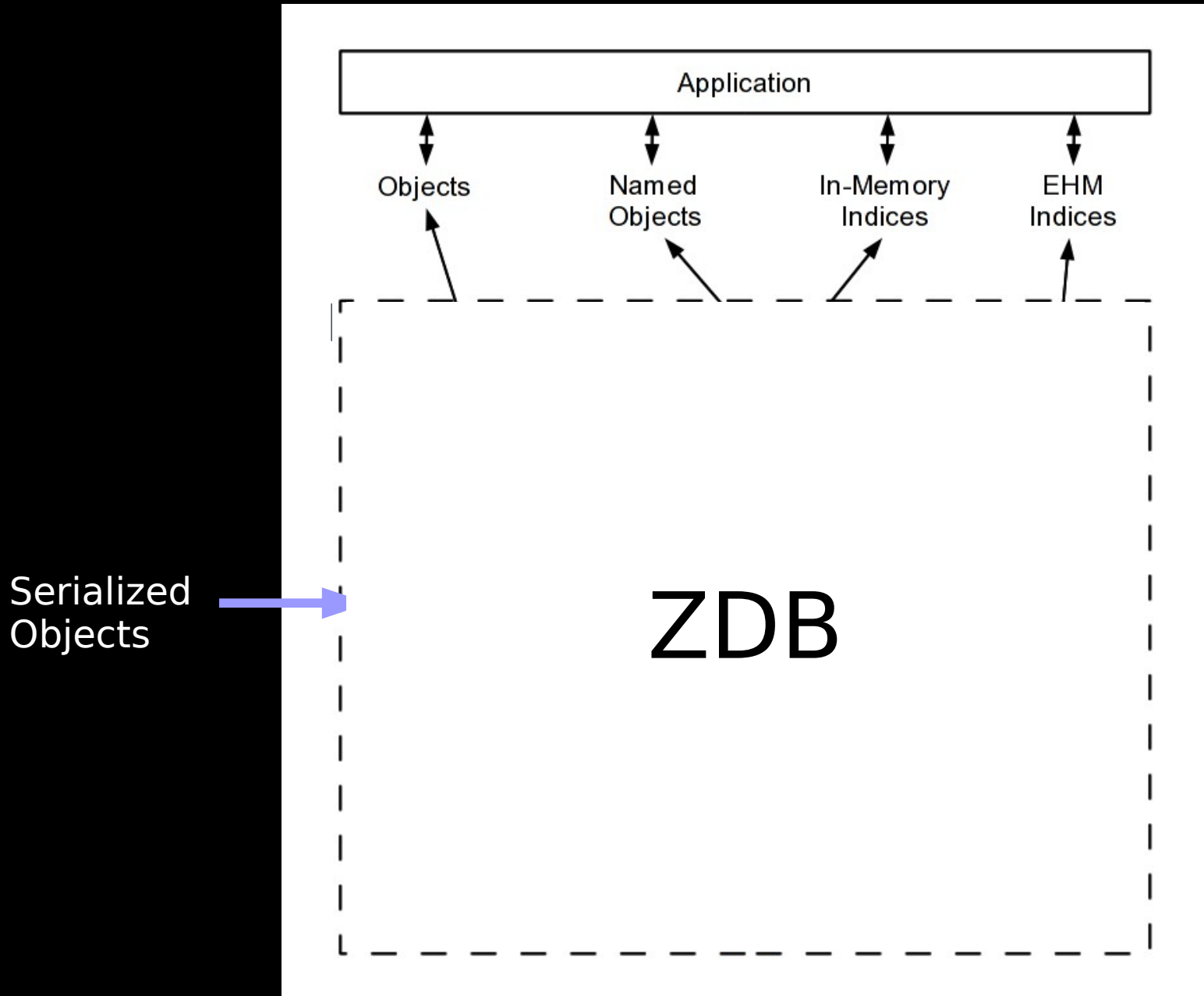


ZDB

ZDB (1/2)

- Object database
- Tuned for extreme EDA performance
 - Creation of objects is fast
 - Retrieval is fast
 - Modification/deletion: slow / unsupported
 - Based on std Java Serialization
- Objects are identified by numbers (DBIDs)
- Offers simple indices: String -> DBID

ZDB



Demo

zamiaCAD Architecture

Eclipse Plugin

Analysis+Navigation

- Signal tracing
- Reference tracing
- Outlines, trees
- Path handling

Waveform + Simulation

- VCD import
- Built-in simulator
- Annotation

Design Entry

- Incremental builder
- Completion
- Error markers

FS Cache

Indexer,
Parser

Syntax Tree

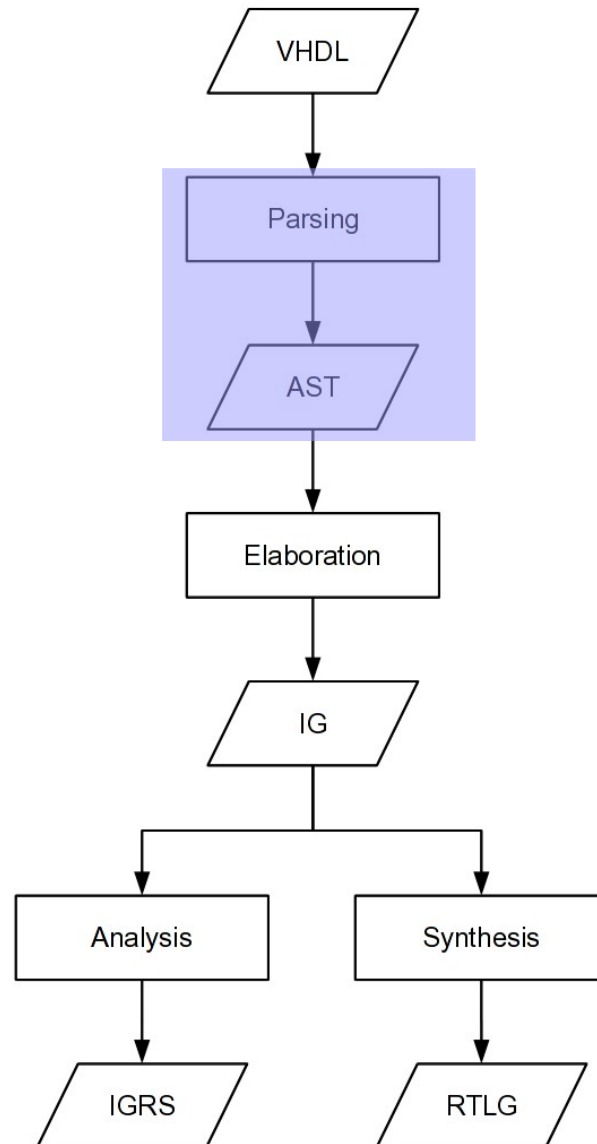
Elaboration
Engine

Model



ZDB

Dataflow



Parser / AST

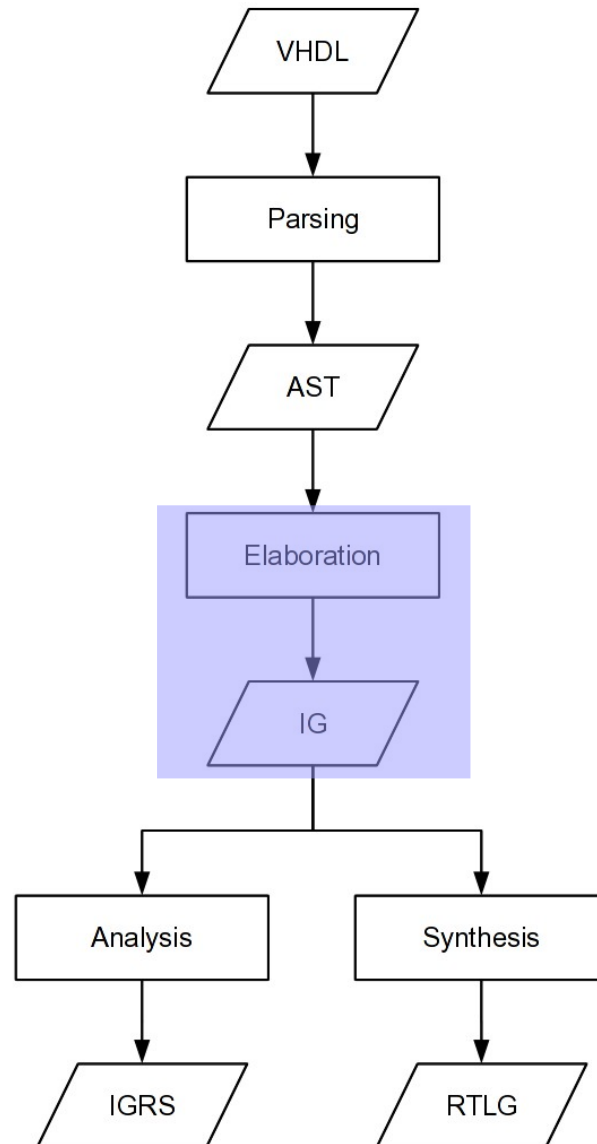
- AST data structures are trees
- Names/Identifiers are not resolved
- Each library unit stored separately in ZDB
 - means each LU has to fit in RAM
 - not true for dumped netlists
 - => concurrent statements are stored separately

Demo

AST Applications

- Editor outline view
- Identifier completion proposals
- Simple identifier occurrence tracing
- Editor folding
- Simple refactorings

Dataflow



IG/Elaboration

- Densely connected graph
 - More difficult to persist in a scalable way
 - Graph-nature is due to resolved names
 - Natural to break up exactly there:
 - anything denoted by names is referred to by DBID
 - Mainly two types of objects:
 - IGOjects (signals, variables, constants and files)
 - IGType

Demo

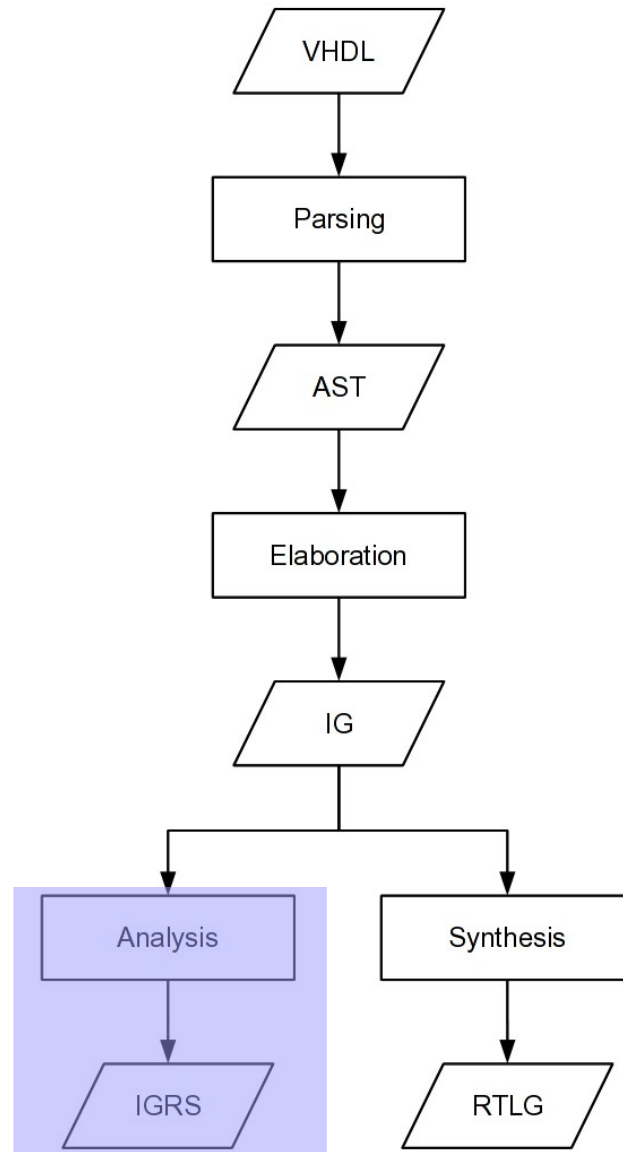
IG Applications (1/2)

- Signal tracing
- Advanced, cross-module refactorings
- Visualization
- Base data structure for further processing
 - Simulation
 - Synthesis
 - Dumping in different languages

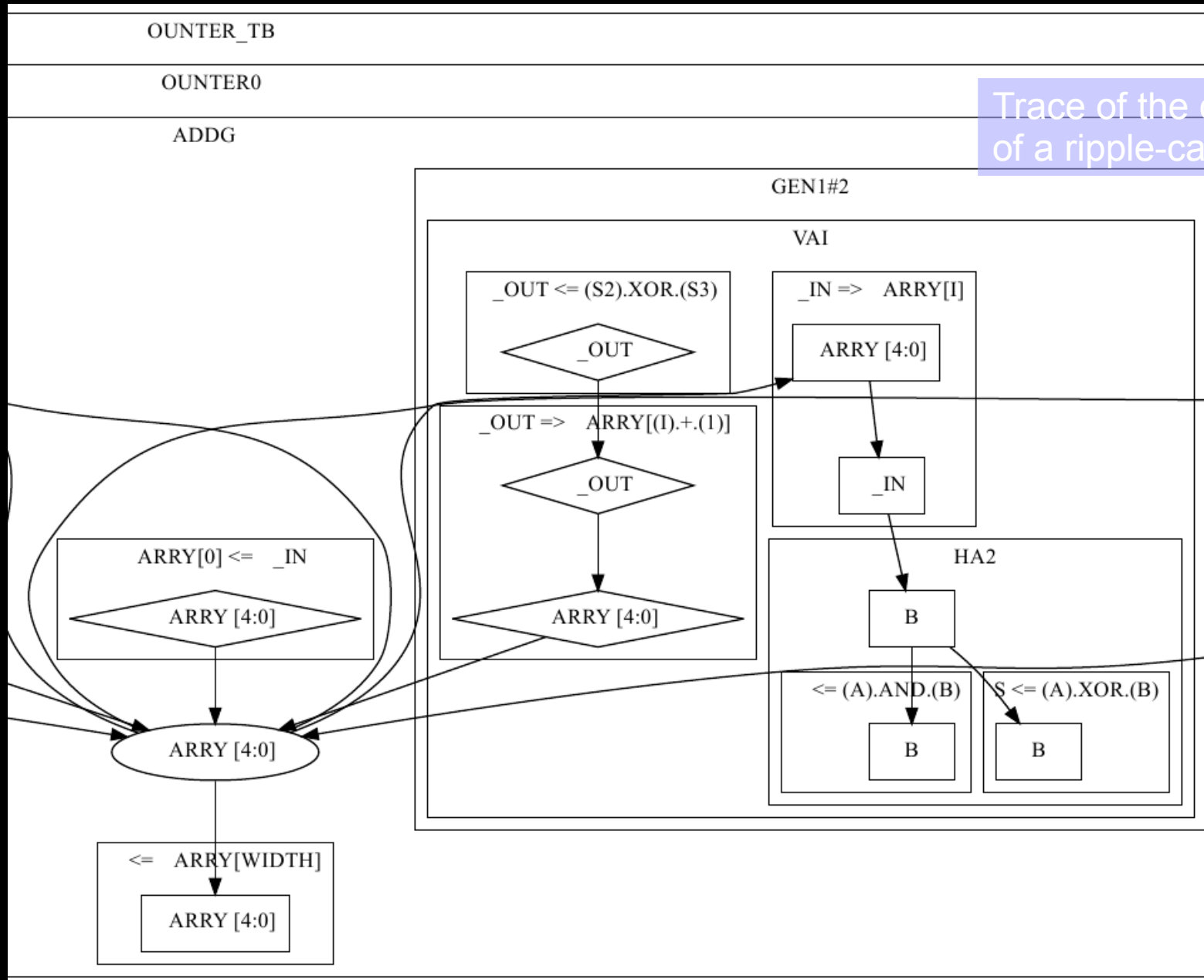
IG Applications (2/2)

- Basic interface to external tools
 - VCD Tools
 - Verification tools
 - Timing annotation
- Most of this data can be matched using hierarchical signal name information

Dataflow

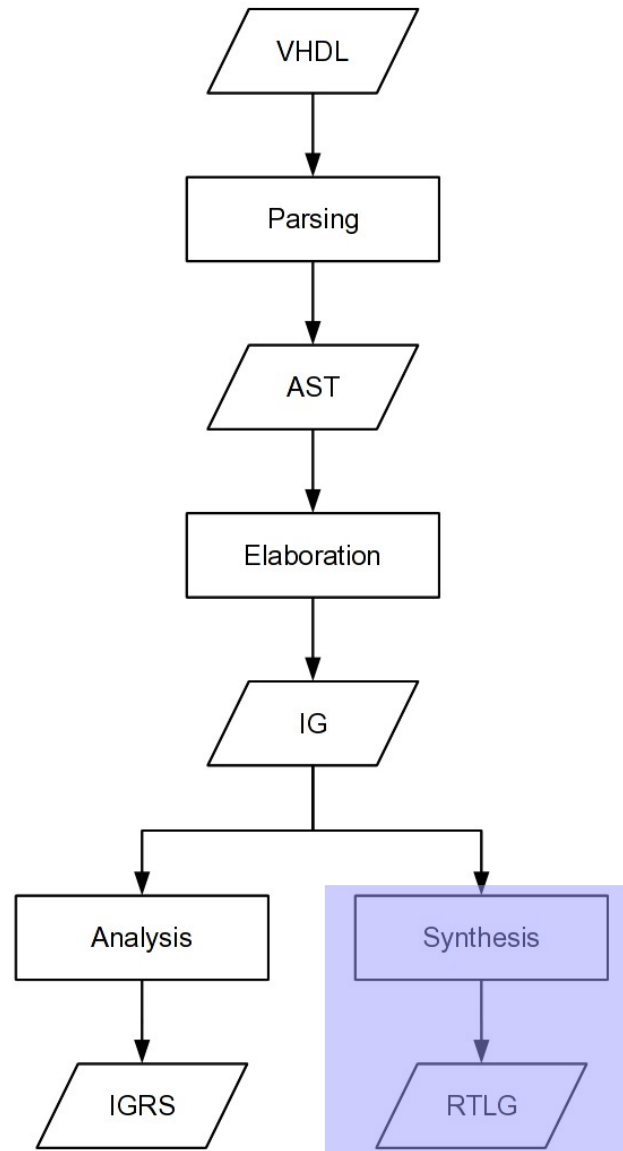


IGRS Example (Excerpt)



Trace of the carry signal of a ripple-carry adder

Dataflow



RTL Synthesis

- Guided by IEEE Std 1076.6
- Map IG to equivalent graph using RTL elements
 - Infer memory
 - Logic and arithmetic modules
 - Deal with data types, busses, ...
- (Very much) work in progress

Demo

Thank You!