Adding VHDL support to Icarus Verilog

Maciej Sumiński, CERN

FOSDEM, Brussels, 1.02.2015

Icarus Verilog

- FOSS hardware description language simulator
- Lead developer: Stephen Williams
- Written in C/C++ (& flex/bison/gperf)
- Great coverage of Verilog (IEEE 1364-1999/2001/2005)
- Active work on SystemVerilog (IEEE 1800-2005/2009/2012)

Goal

- GHDL = VHDL simulator
- Icarus = Verilog & SystemVerilog simulator
- ? = mixed-mode simulator

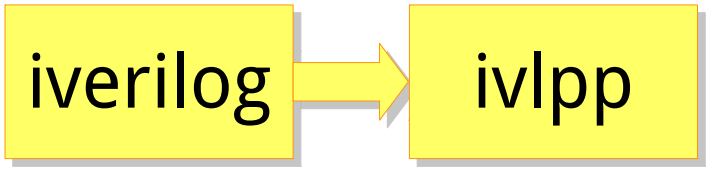
Icarus principles

```
iverilog adder.vhd adder_test.v
```

iverilog

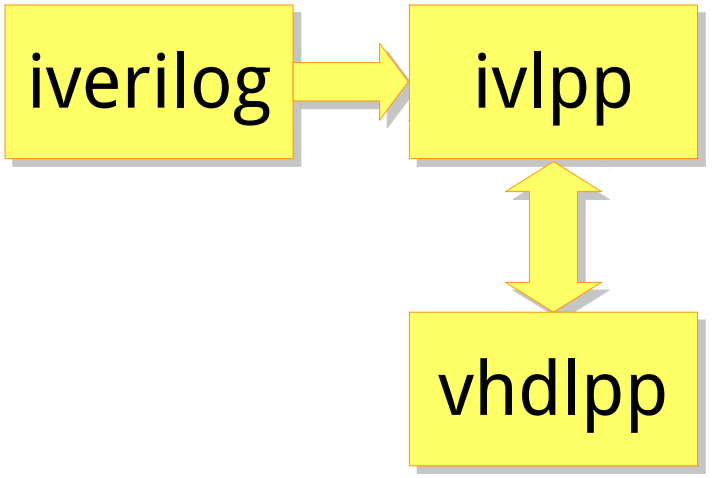
Icarus principles

```
iverilog adder.vhd adder_test.v
```



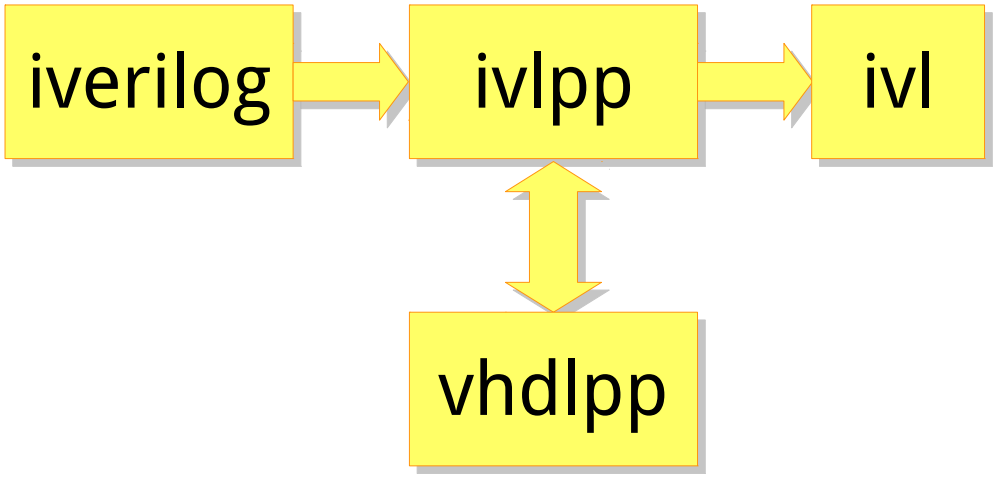
Icarus principles

```
iverilog adder.vhd adder_test.v
```



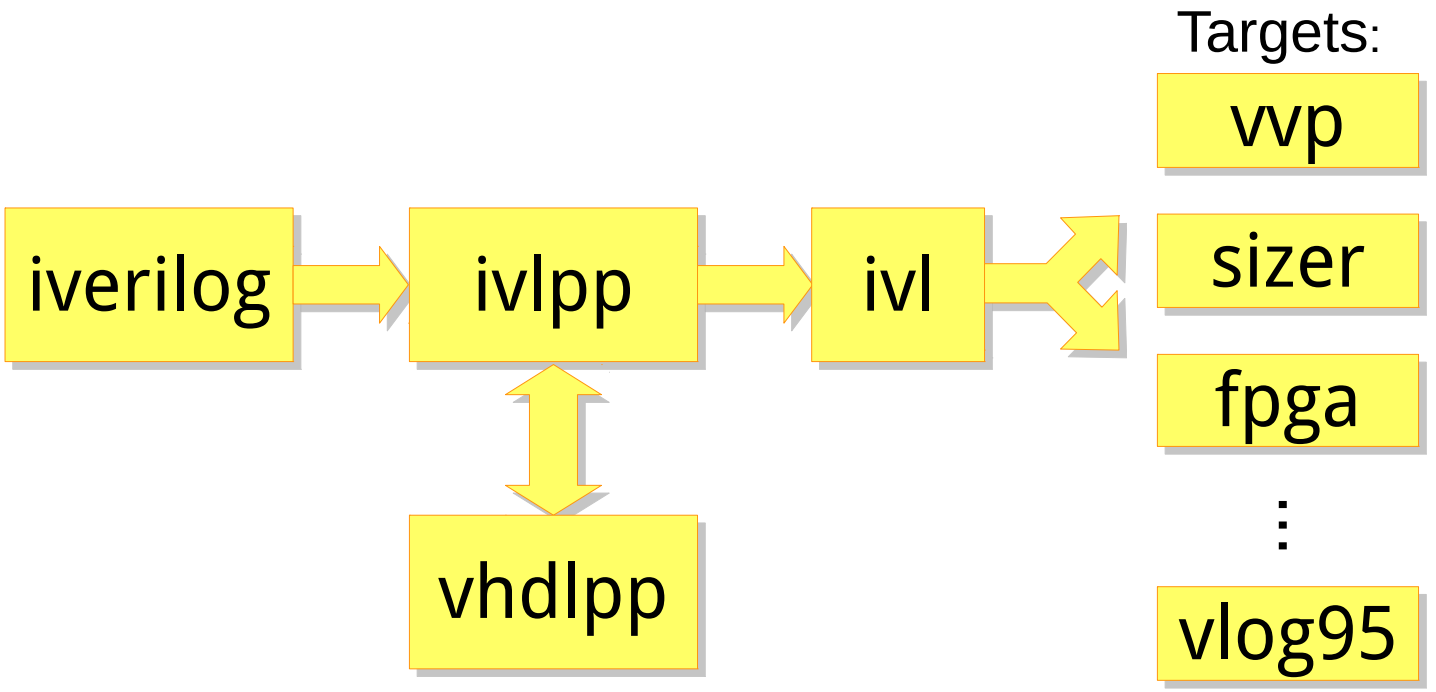
Icarus principles

```
iverilog adder.vhd adder_test.v
```



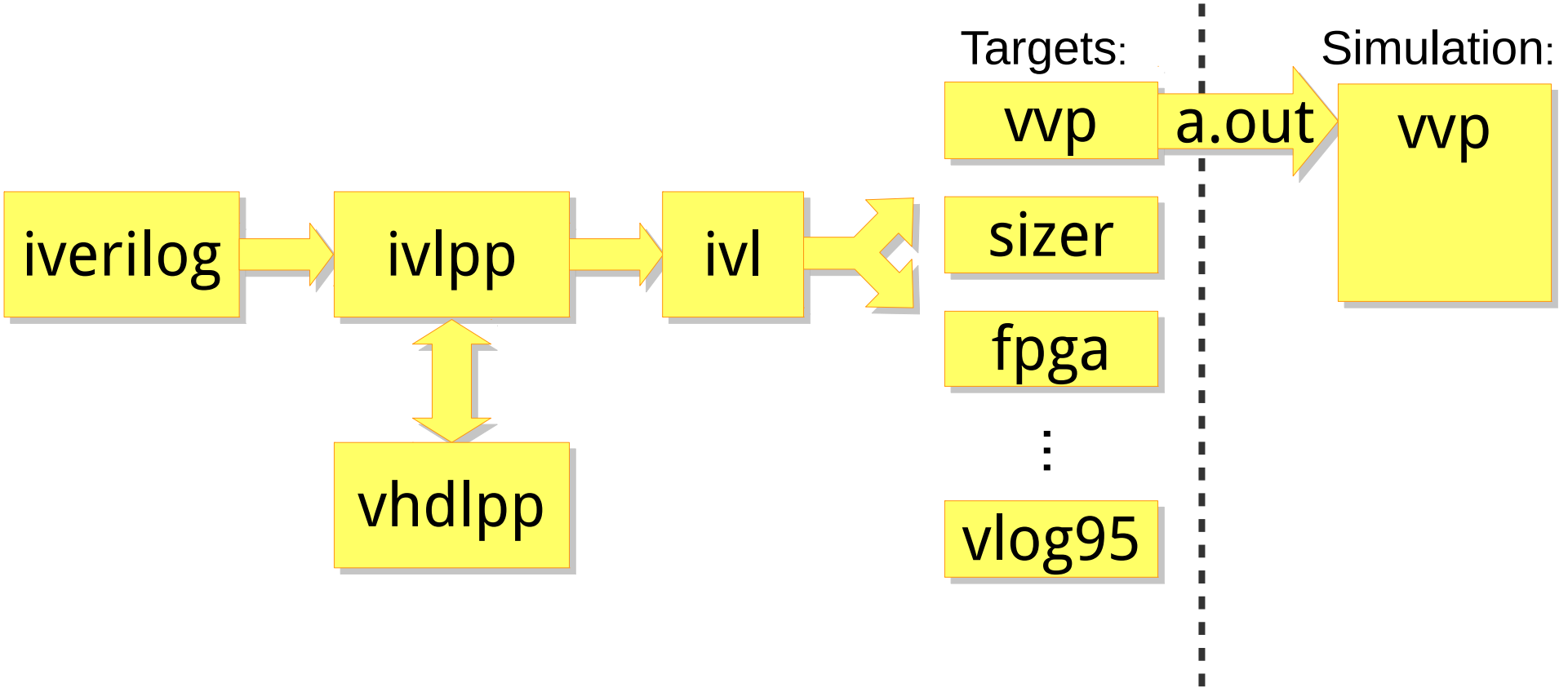
Icarus principles

```
iverilog adder.vhd adder_test.v
```



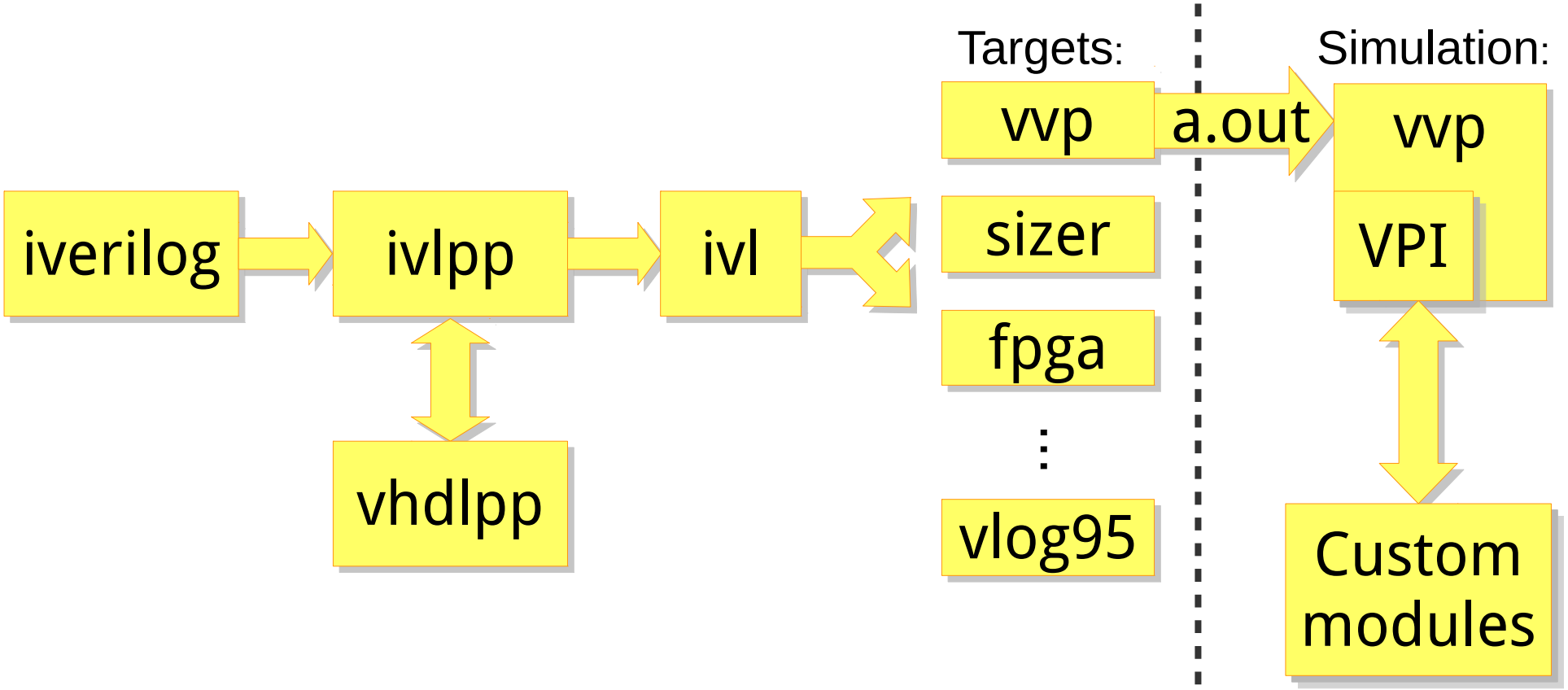
Icarus principles

```
iverilog adder.vhd adder_test.v
```



Icarus principles

```
iverilog adder.vhd adder_test.v
```



vhdlpp - example

```
library ieee;
use ieee.std_logic_1164.all;

entity mux2to1 is
  port(
    i0, i1, s: in std_logic;
    y: out std_logic);
end mux2to1;

architecture mux2to1_rtl of
mux2to1 is
begin
  process (i0, i1, s)
    begin
      case (s) is
        when '0' => y <= i0;
        when others => y <= i1;
      end case;
  end process;
end mux2to1_rtl;
```

```
module \mux2to1 (input wire
logic \i0  ,
input wire logic \i1  ,
input wire logic \s  ,
output logic \y  );
always begin
case (\s )
1'b0:
\y  <= \i0 ;
default:
\y  <= \i1 ;
endcase
@(\i0 , \i1 , \s ) /*
sensitivity list for process
*/;
end
endmodule
```

vhdlpp

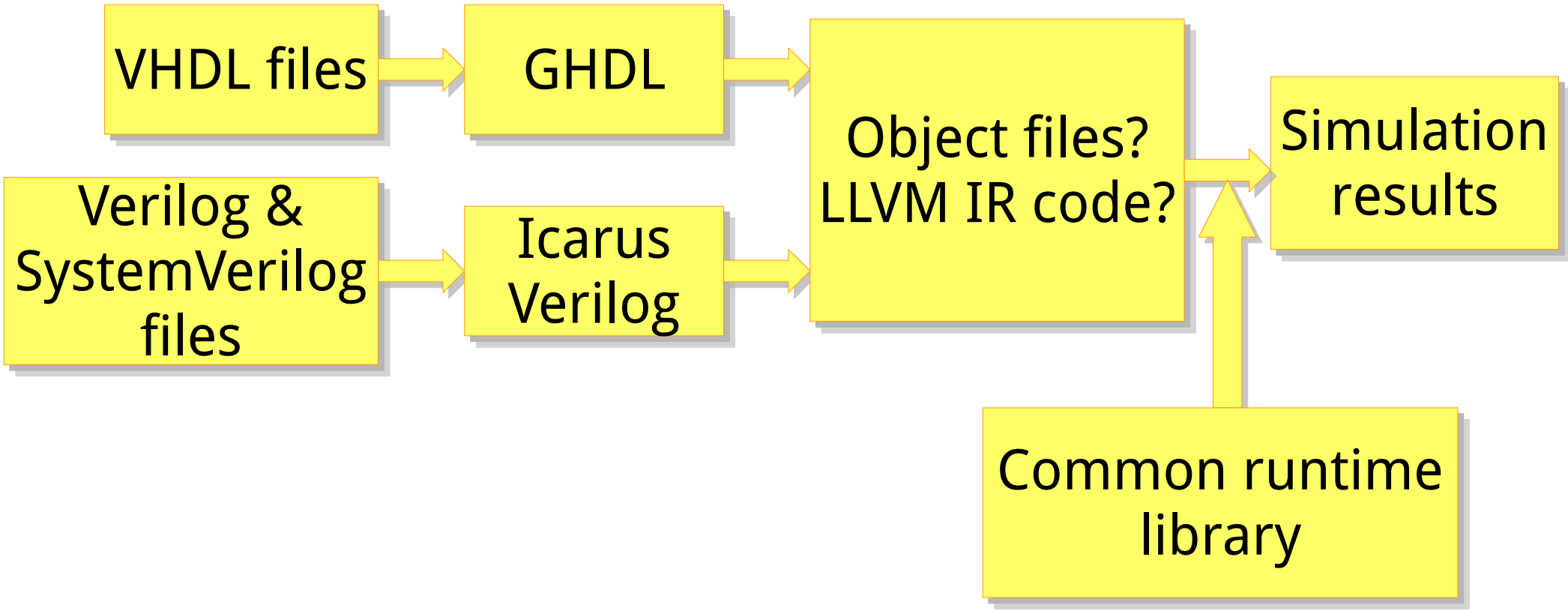
Adding new features:

- Parser rules
- Elaborate
- Emit

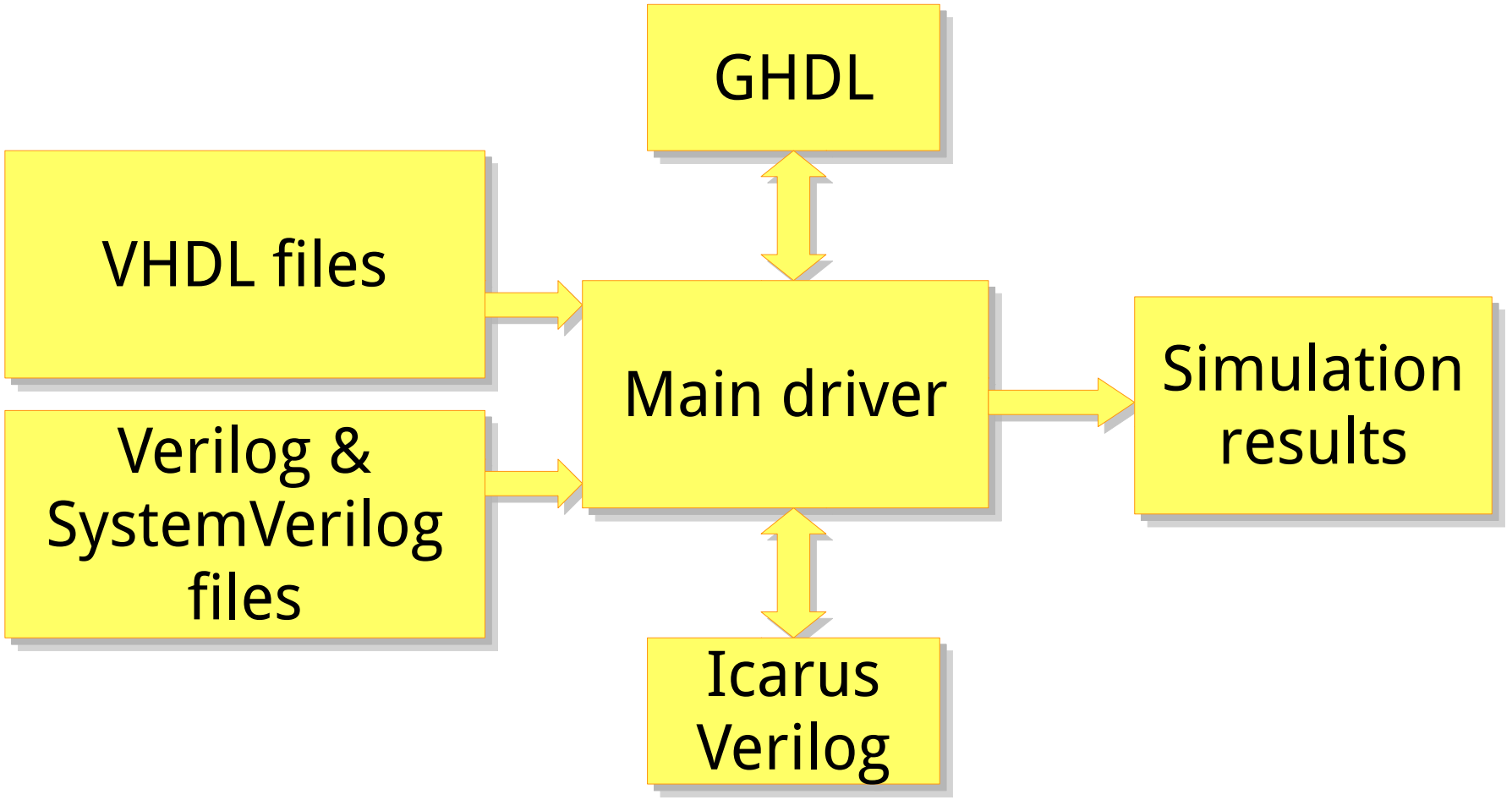
Status

- Procedures & functions
- Loops, including for .. generate
- Typedefs / subtypes
- Arrays, records
- Some of attributes (e.g. 'event, 'range)
- 80+ tests for VHDL & over 2000 for Verilog & SV

Alternative approach



Another method



More information

- Official website

  http://iverilog.icarus.com/

- Wiki

  http://iverilog.wikia.com/

- Github repository:

  https://github.com/steveicarus/iverilog/

  https://github.com/steveicarus/ivtest/

In a Nutshell, Icarus Verilog...

...has had 7,561 commits made by 36 contributors

representing 175,756 lines of code

...is mostly written in C++

with an average number of source code comments

...has a well established, mature codebase

maintained by a large development team

with increasing Y-O-Y commits

...took an estimated 46 years of effort (COCOMO model)

starting with its first commit in November, 1998

ending with its most recent commit 22 days ago

[source: https://www.openhub.net/p/iverilog]