Plain and Simple HDL

Creating the Arduino for FPGAs

Twitter: @PSHDL

Motivation

- VHDL and Verilog are hard to learn
- Synthesis tools are hard to use
- It’s no fun to learn programming FPGAs

Twitter: @PSHDL

PSHDL primary goals

- Zero installation
- Easy to learn
- Inviting and user friendly development environment
- A powerful platform, not just a new language
- A collaborative community

Twitter: @PSHDL

Browser as IDE?!

- Zero installation
- Powerful editing
  - Syntax highlighting
  - Error marking on save
- Easy collaboration

Twitter: @PSHDL

Demo time!

Twitter: @PSHDL

Browser as IDE!

- Simple simulation
  - still faster than iSim in many cases
- User provided extension via JavaScript
- Try it at: http://beta.pshdl.org

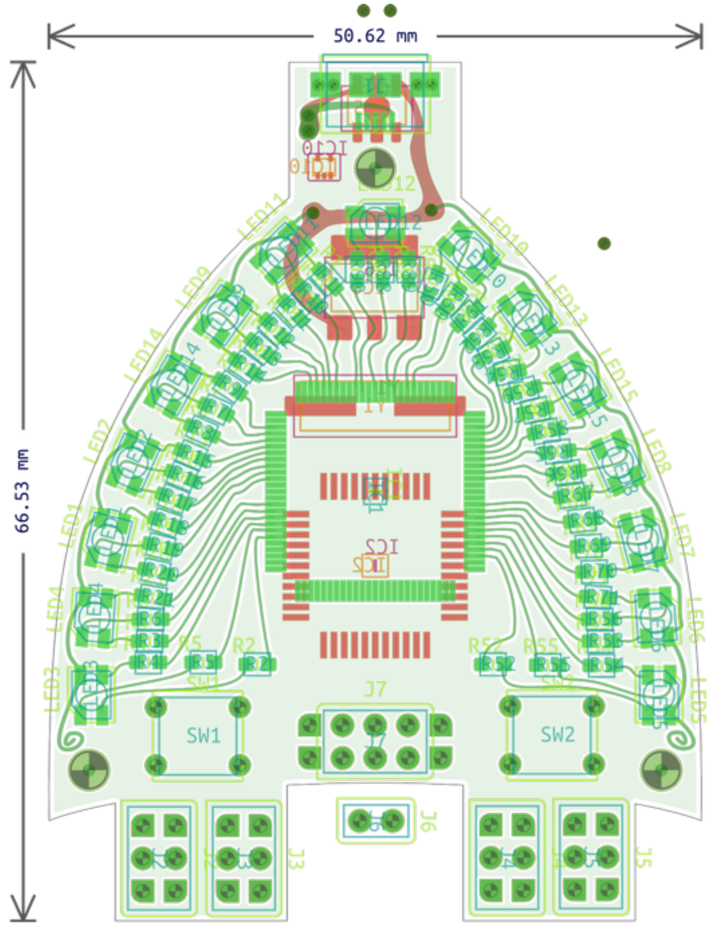
Twitter: @PSHDL

Create a new HW platform

- Cheap (30 € or less)
- Fun (blinking RGB LEDs)
- Challenging (networking, SoC)
- Extensible (different arms)

Twitter: @PSHDL

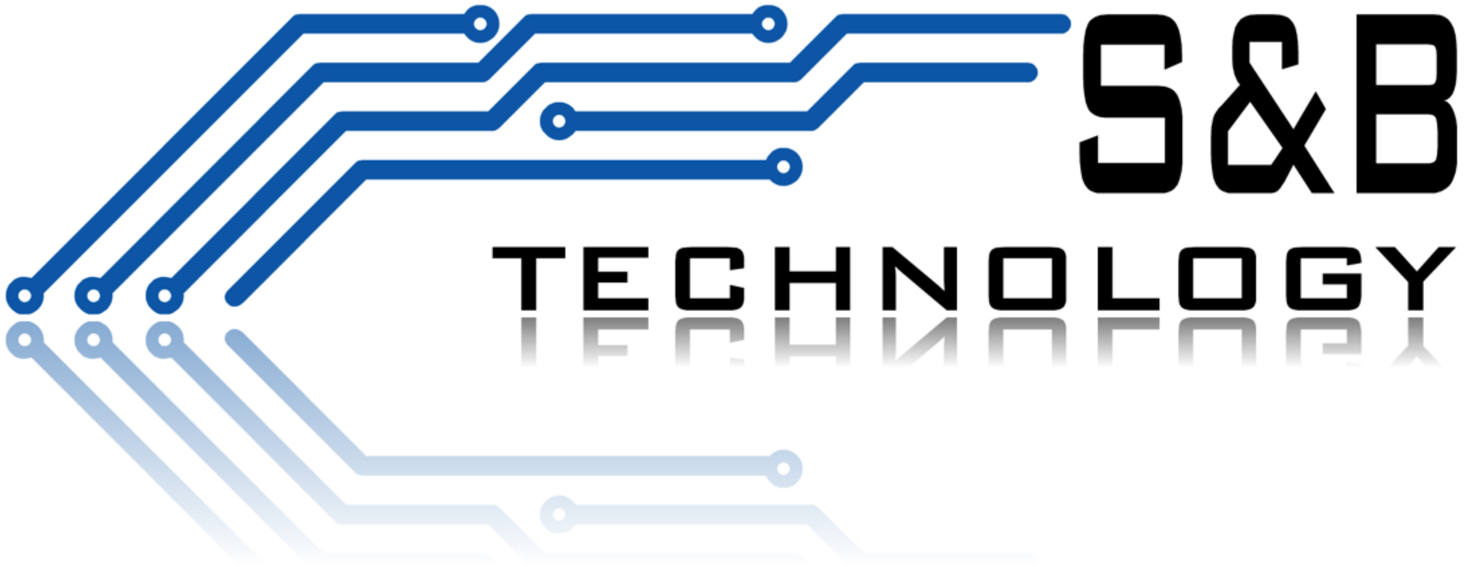
First design



boldport

Twitter: @PSHDL

58 drills: 8 3 11 36



@PSHDL

HW Platform

- Actel A3PN250 VQ100
- Atmel XMega 32A4U
  - Programming from any platform
  - High speed USB connection to PC
  - Additional IOs (SoC possible)

Twitter: @PSHDL

Connectivity

- 4 Ports (16 Pin IDC header, 14 IO)
- 2 Vertical ports (4 Pin IDC header, 2 IO)
- 5 IO to Atmel XMega (for example SPI, UART, Clock)
- 4 push buttons
- Total of 60 user accesible IOs with 3.3v
- Networking:
  - 4 horizontal UART via arm
  - 2 vertical UART via 45° stacking

Twitter: @PSHDL

Default Arm

- 16 Pin IDC header
  - 12 connect to 4 RGB LEDs
  - 2 connect to 4 Pin IDC header at end

Availability

- Plan to sell it for 3x €
  - x depends on how many we could sell
  - hopefully x<=0
- If you are interested => boards.pshdl.org

Twitter: @PSHDL

Demo Time

Twitter: @PSHDL

PSHDL Target groups

- First time FPGA users
  - Low barrier of entrance
    - Get a blinking LED in less than a minute
- Hackers/Makers
  - Extensible platform
  - Cheap
  - **Open Source Tools**

Twitter: @PSHDL

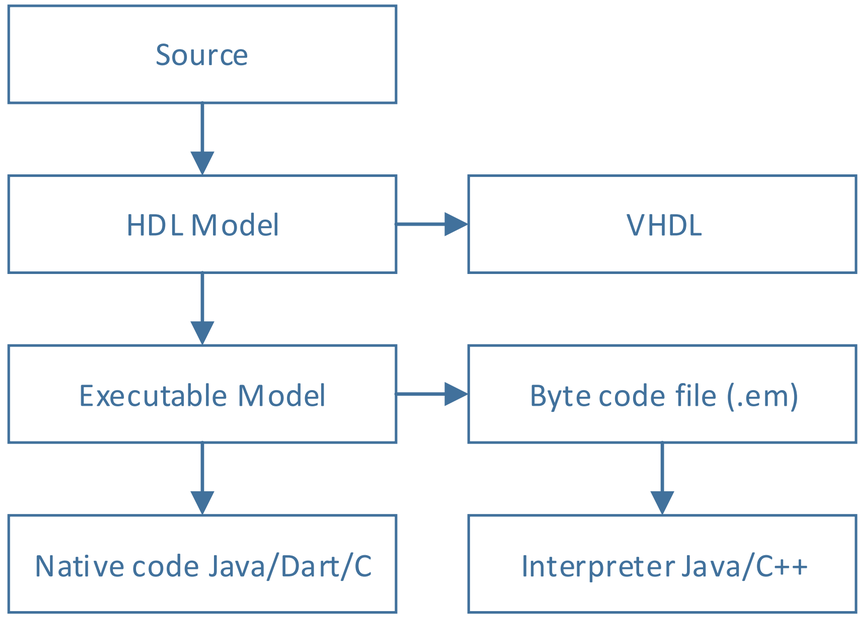
PSHDL for experts

- Embeddable Simulation
  - Multi-language support
    - C/C++/Java/Dart
  - Unbelievably fast
  - Platform neutral byte code representation
- Embeddable Domain Specific Languages

Twitter: @PSHDL

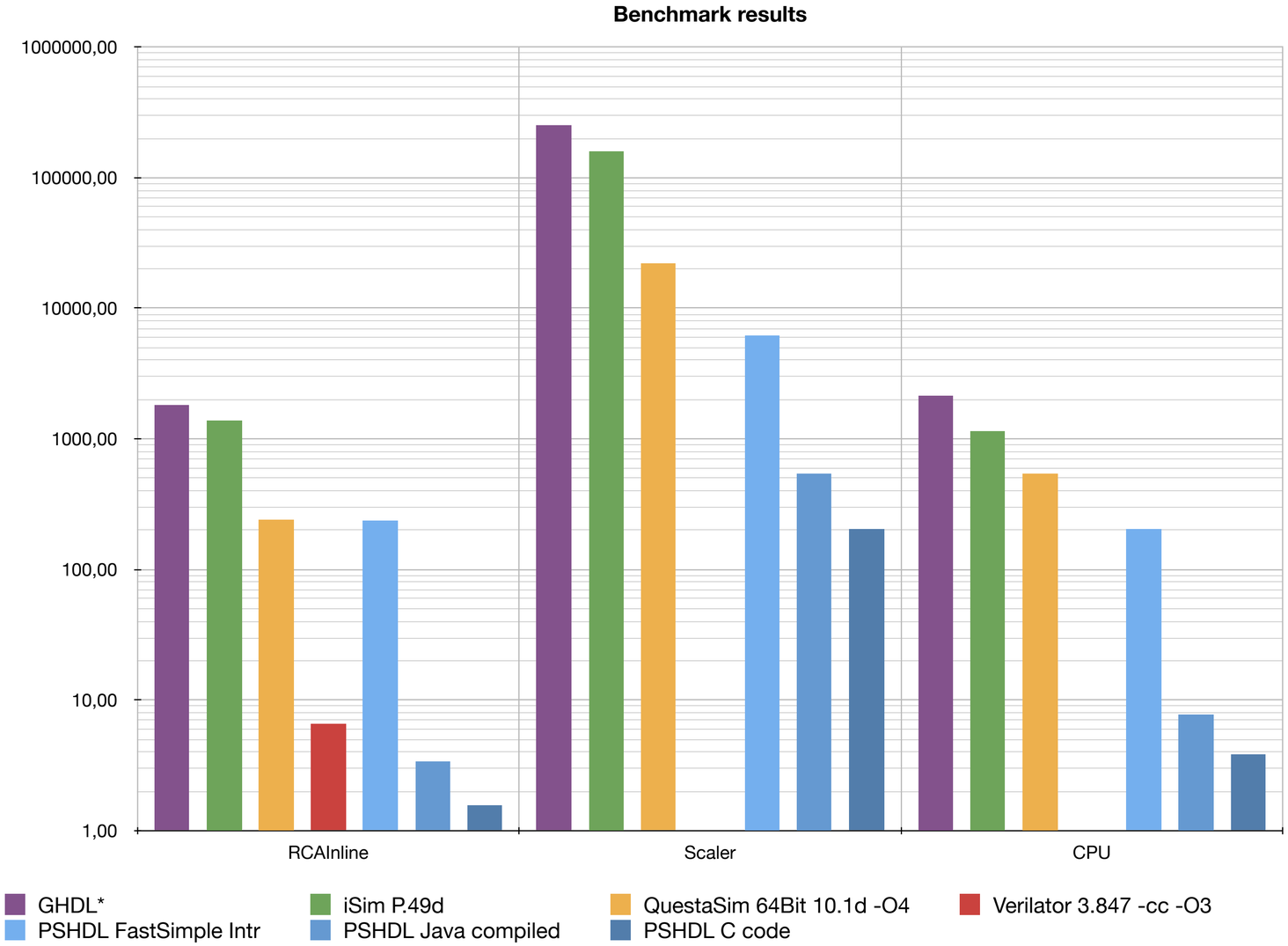
Code generation overview

- HDL Model
  - Good for transformation
- Executable Model
  - Sequential data flow



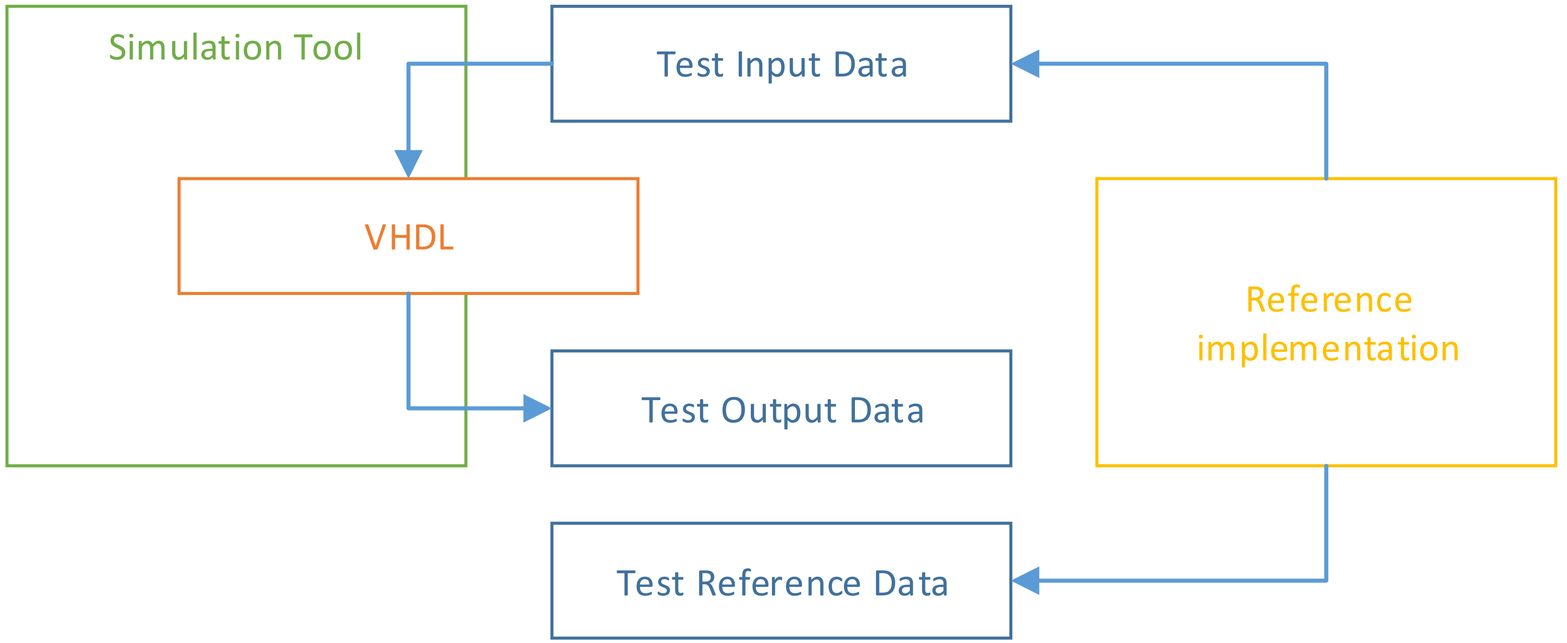
Twitter: @PSHDL

Simulation



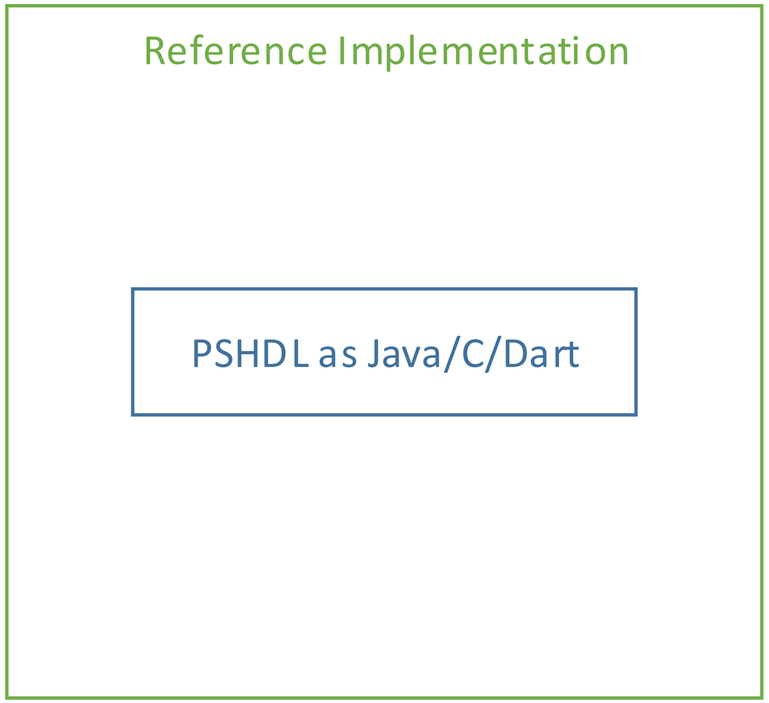
Twitter: @PSHDL

VHDL Simulation



Twitter: @PSHDL

PSHDL Simulation



Twitter: @PSHDL

Example DSL

- Problem: Develop an IP Core attached to CPU
- Memory mapped IO access to registers

PSHDL Generators

```
module BUSTest{
  include Bus bus=generate plb()<[
    row input{
      rw register uint<16> a;
      rw register uint<16> b;
    }
    row output {
      fill;
      r register uint<16> result;
    }
    column adder {
      input;
      output;
    }
    memory {
      adder[4];
    }
]>;
  for (i={0:3}){
    bus.result[i]=bus.a[i]+bus.b[i];
  }
}
```

Offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Row
adder [0]																																	
0 [0x00]	*a [0]*																*b [0]*															input [0]
4 [0x04]	unused																*result [0]*															output [0]
adder [1]																																	
8 [0x08]	*a [1]*																*b [1]*															input [1]
12 [0x0c]	unused																*result [1]*															output [1]
adder [2]																																	
16 [0x10]	*a [2]*																*b [2]*															input [2]
20 [0x14]	unused																*result [2]*															output [2]
adder [3]																																	
24 [0x18]	*a [3]*																*b [3]*															input [3]
28 [0x1c]	unused																*result [3]*															output [3]

The generated Interface looks like this:

```
interface Bus{
   in register uint<16> result[4];
   inout register uint<16> b[4];
   inout register uint<16> a[4];
}
```

Twitter: @PSHDL

PSHDL Generators

```
#ifndef BusDefinitions_h
#define BusDefinitions_h

#include "BusStdDefinitions.h"

//Typedef
typedef struct input {
    bus_uint16_t    a;
    bus_uint16_t    b;
} input_t;
// Setter
int setInputDirect(uint32_t *base, int index, bus_uint16_t a, bus_uint16_t b);
int setInput(uint32_t *base, int index, input_t *newVal);
//Getter
int getInputDirect(uint32_t *base, int index, bus_uint16_t *a, bus_uint16_t *b);
int getInput(uint32_t *base, int index, input_t *result);
//Typedef
typedef struct output {
    bus_uint16_t    result;
} output_t;
//Getter
int getOutputDirect(uint32_t *base, int index, bus_uint16_t *result);
int getOutput(uint32_t *base, int index, output_t *result);
typedef struct adder {
    input_t input;
    output_t output;
} adder_t;

#endif
```

Twitter: @PSHDL

Demo

Twitter: @PSHDL

Thanks, Questions?

Check out:
beta.pshdl.org (new web interface)
boards.pshdl.org (new board)
blog.pshdl.org (blog with news)
code.pshdl.org (source code)
twitter: @PSHDL
facebook: PSHDL
G+: +PshdlOrg