TALLINN UNIVERSITY OF TECHNOLOGY

Pyha - Object-Oriented Hardware Description Language based on Python

Gaspar Karm

Master's Thesis

Thomas Johann Seebeck Department of Electronics

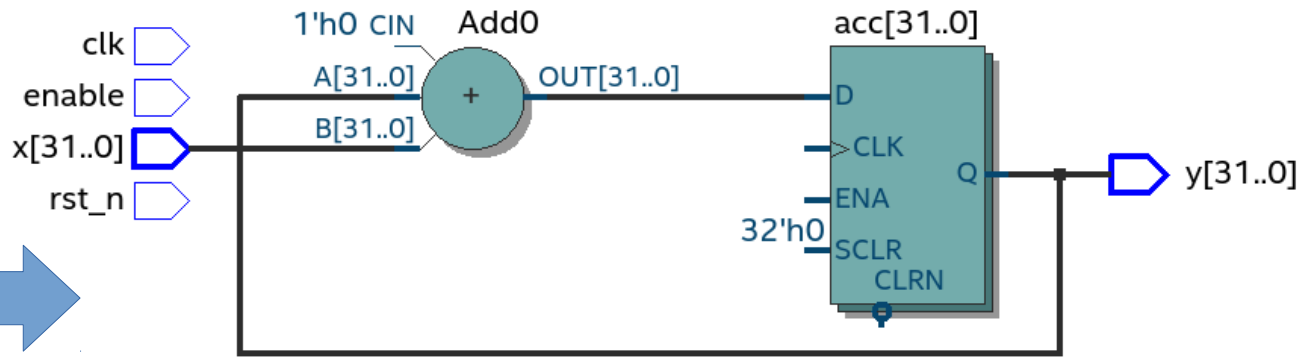
06.06.2017

Background

VHDL - Hardware description language

```
entity Accumulator is
    port (
        clk, rst_n, enable: in std_logic;
        x: in std_logic_vector(31 downto 0);
        y: out std_logic_vector(31 downto 0)
    );
end entity;

architecture arch of Accumulator is
   signal acc: integer := 0;
begin
   process(clk, rst_n)
   begin
      if (not rst_n) then
         acc <= 0;
      elsif rising_edge(clk) then
         if enable then
            acc <= acc + to_integer(signed(x));
         end if;
      end if;
      y <= std_logic_vector(to_signed(acc, 32));
   end process;
end architecture;
```

Synthesised logic (Quartus RTL viewer)

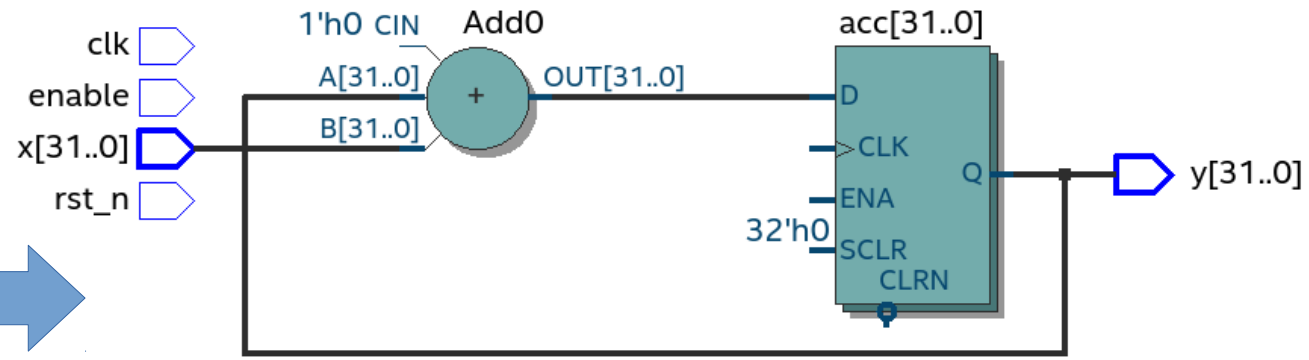


Source: https://en.wikipedia.org/wiki/Field-programmable_gate_array

My work

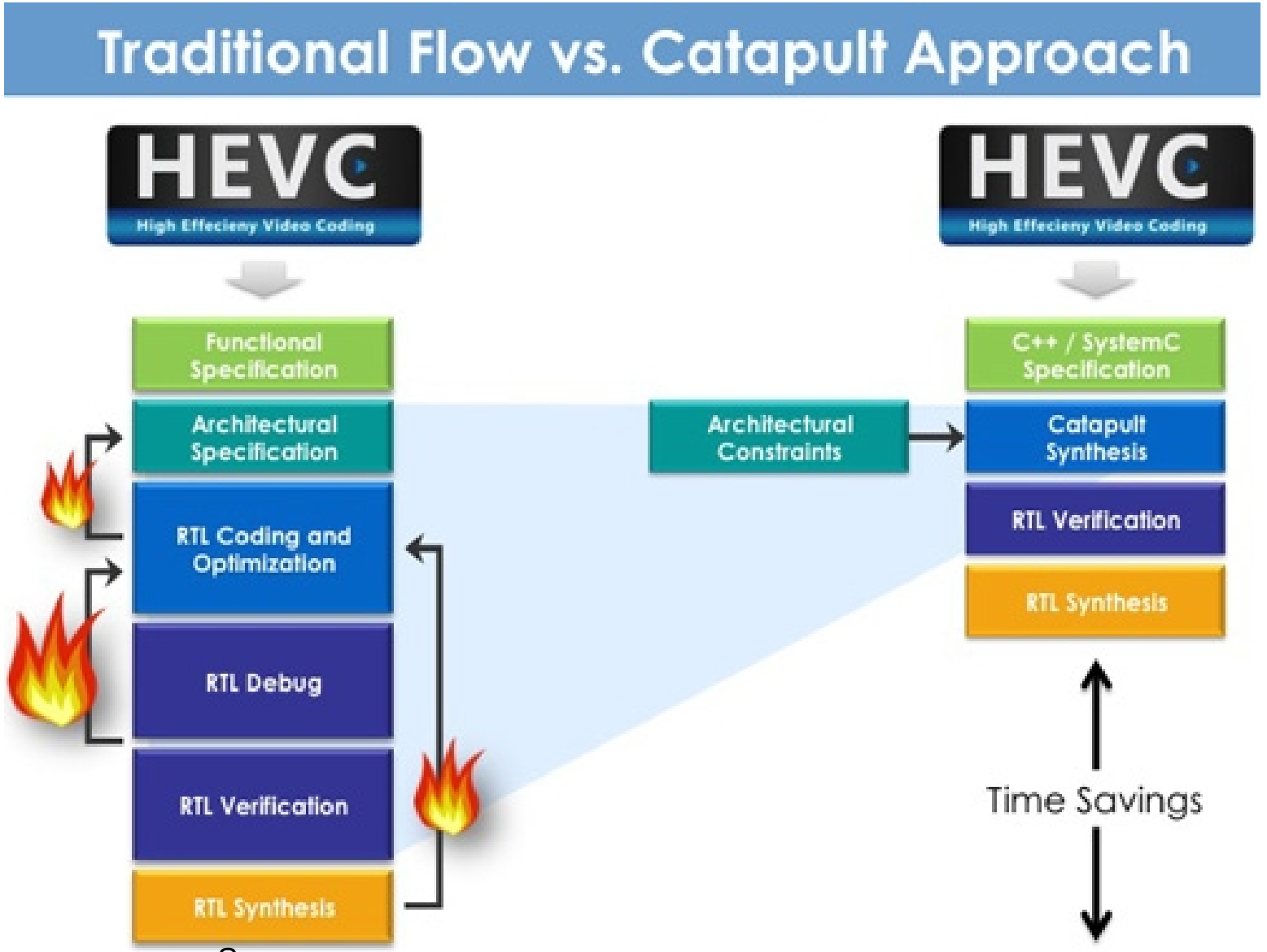
```
class Accumulator(HW):
    def __init__(self):
        self.acc = 0

    def main(self, x):
        self.acc = self.acc + x
        return self.acc
```

Synthesised logic (Quartus RTL viewer)



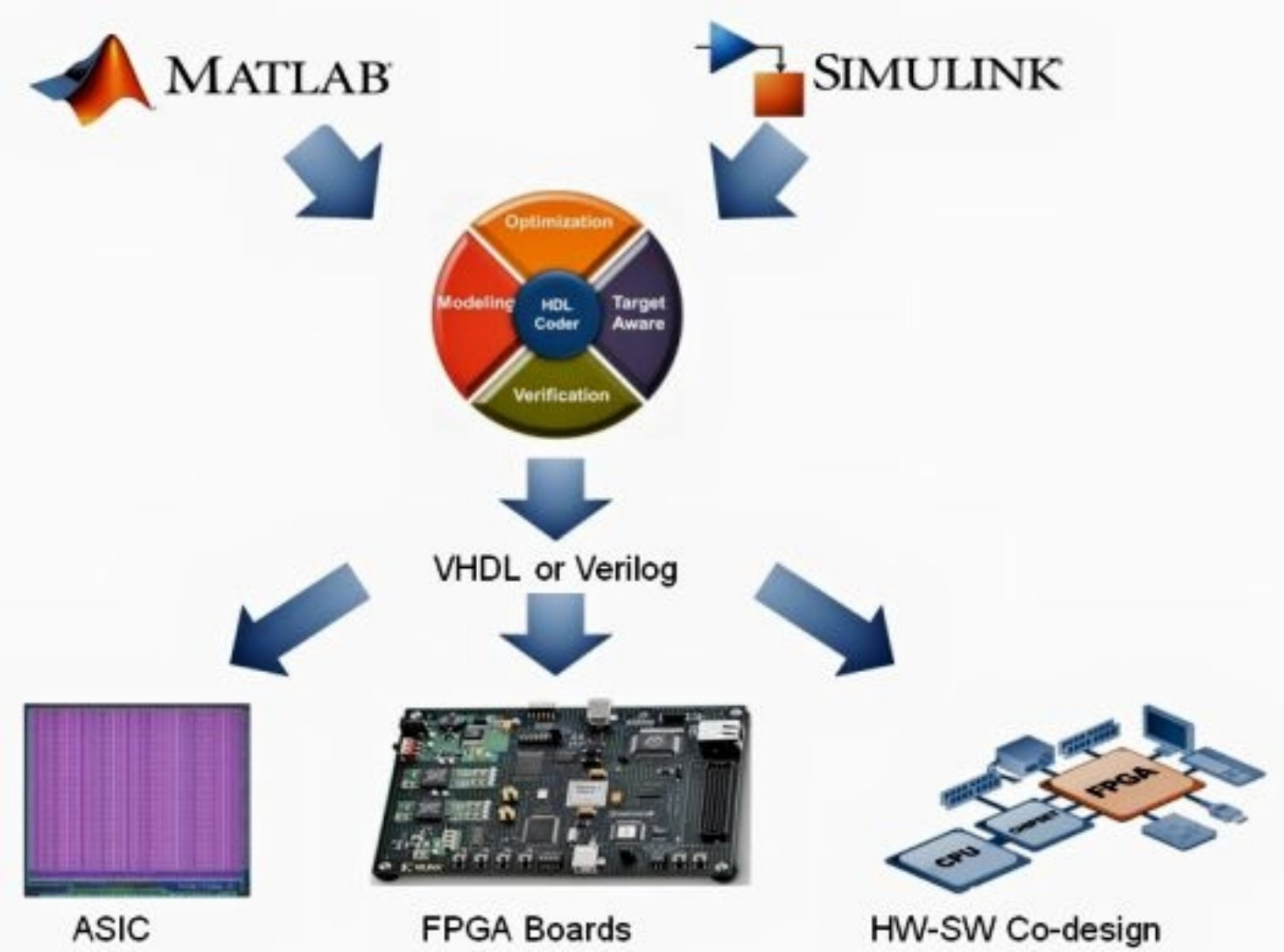
Source: https://en.wikipedia.org/wiki/Field-programmable_gate_array

Related work - High Level Synthesis (HLS)



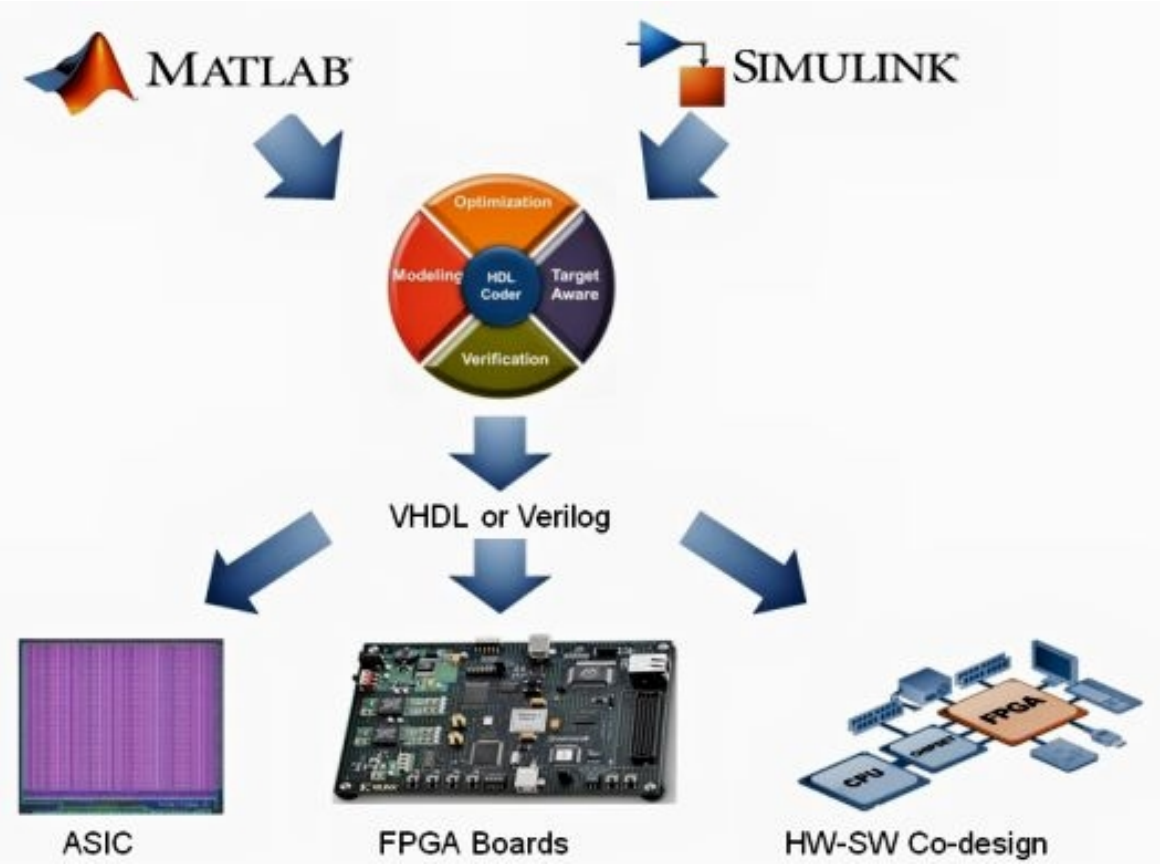
Source:
https://www.semiwiki.com/forum/content/2652-when-good-time-start-using-high-level-synthesis.html

MATLAB as HLS



Source:
https://forums.xilinx.com/t5/Xcell-Daily-Blog/MathWorks-HDL-Coder-wins-Embedded-World-AWARD-in-Nuremberg-last/ba-p/756199

MATLAB as HLS



Source:
https://forums.xilinx.com/t5/Xcell-Daily-Blog/MathWorks-HDL-Coder-wins-Embedded-World-AWARD-in-Nuremberg-last/ba-p/756199

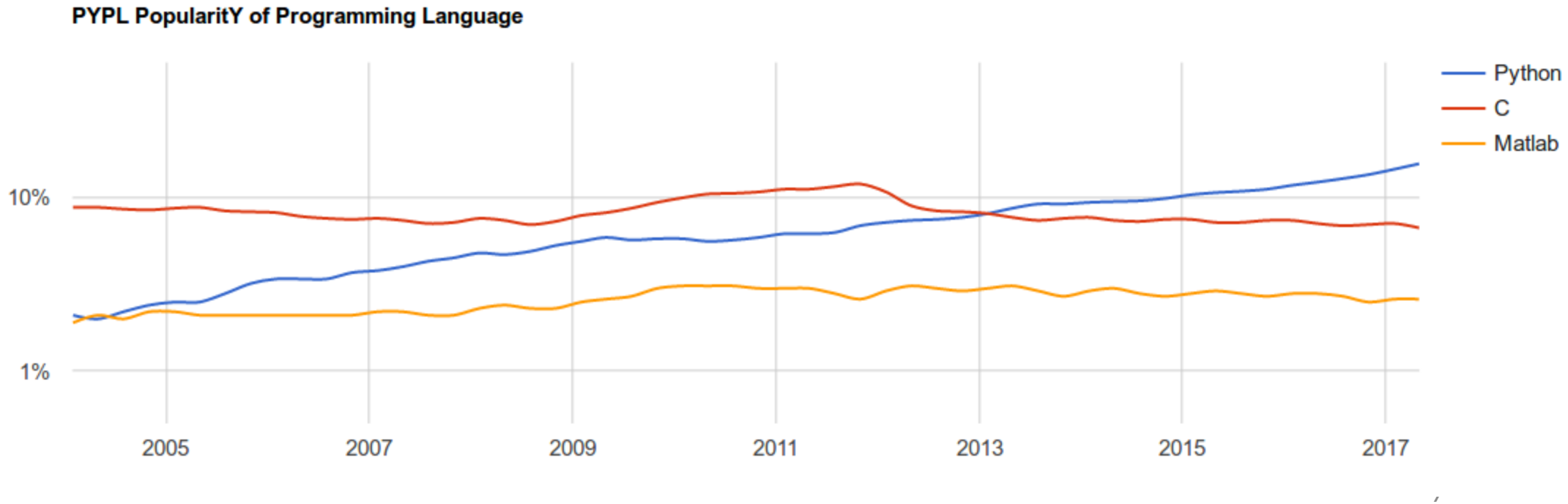
MATLAB	2000 EUR
SIMULINK	3000 EUR
HDLCoder	10000 EUR
HDLVerifier	3500 EUR
Fixed-point toolbox	10000 EUR
DSP System Toolbox	1000 EUR
Signal Processing Toolbox	1000 EUR
DSPBuilder	2000 EUR (annually)
ModelSim	2000 EUR
Quartus	4000 EUR (annually)

Total = ~40 000 EUR

Source:
http://www.eetimes.com/document.asp?doc_id=1317035
https://www.altera.com/buy/design-software.html
https://se.mathworks.com/pricing-licensing.htm

Use Python instead?

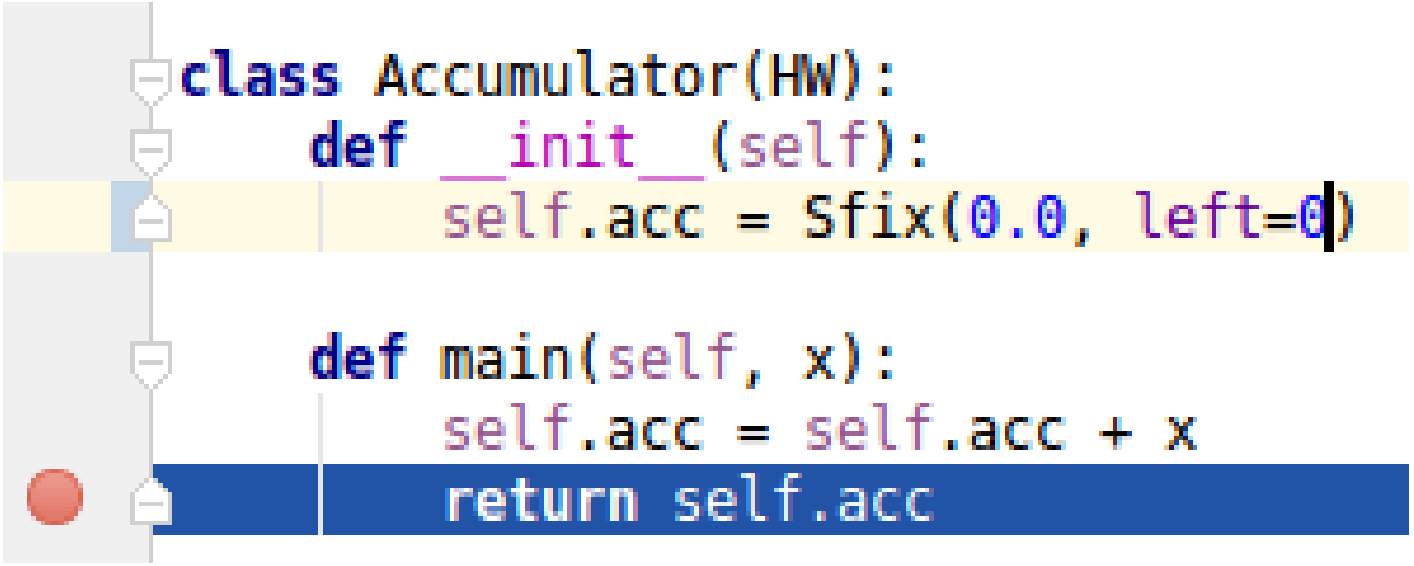
Worldwide, Java is the most popular language, Python grew the most in the last 5 years (8.2%) and PHP lost the most (-4.8%)



Source: http://pypl.github.io/PYPL.html

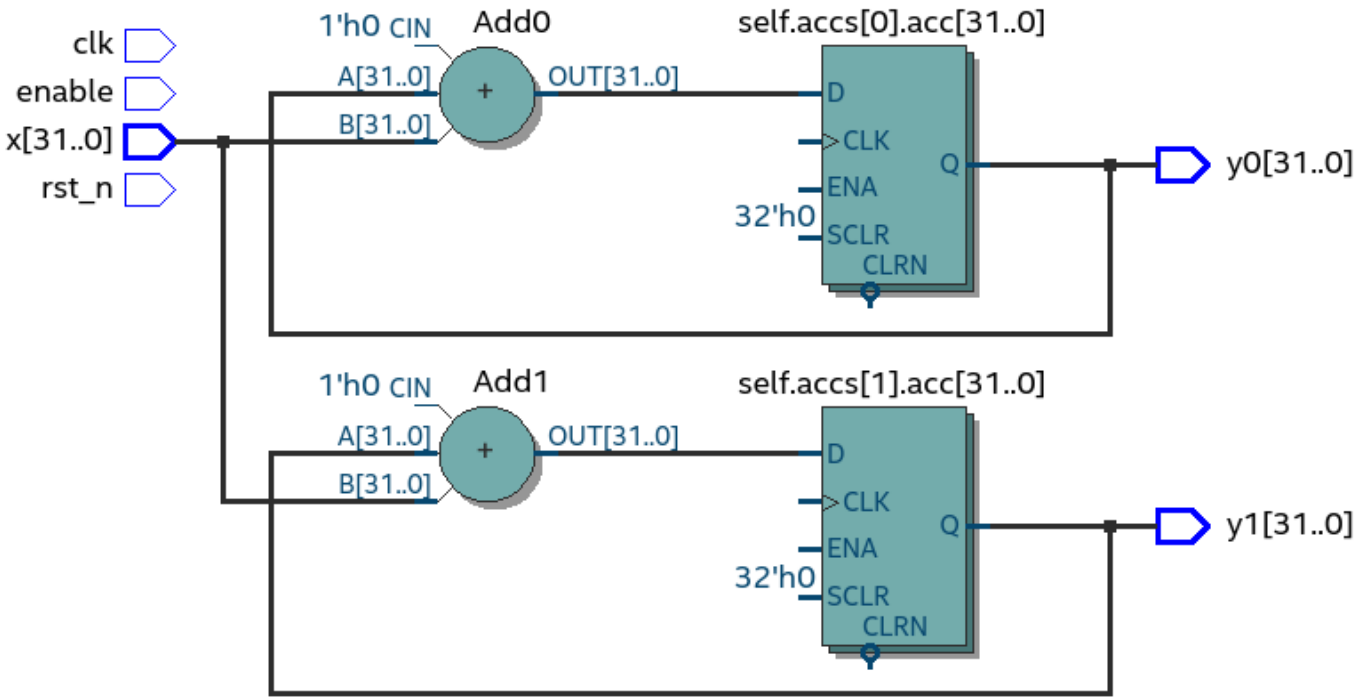
Main features

- Fixed-point support, lazy bounds
- Interpretation/Debug modes:
  - Hardware
  - Model

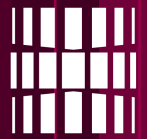


Main Features

- Object orientation
- Pure Python
- Fully sequential execution
- Debuggable




```
class Accumulator(HW):
    def __init__(self):
        self.acc = 0

    def main(self, x):
        self.acc = self.acc + x
        return self.acc


class TwoAccumulators(HW):
    def __init__(self):
        self.accs = [Accumulator(), Accumulator()]

    def main(self, x):  self: <main.TwoAccumulators object at 0x7fcd954c2e10>  x: 3
        y0 = self.accs[0].main(x)  y0: 4
        y1 = self.accs[1].main(x)  y1: 4
        return y0, y1
```



Contributions

- Hardware description, simulation and debugging in Python
- Fixed-point arithmetic library for Python
- Simplified verification
- Sequential object-oriented VHDL model
- Simple method for converting Python to another language
- IP cores

Thank you for listening!

The meaning of PYHA? Is it an acronym?

The meaning of PYHA? Is it an acronym?

PYthon **HA**rdware

What exactly is OOP VHDL model?
Is the OOP-style VHDL thesis author's contribution?
The attempts to enhance OO features of VHDL has been done already 20 years ago.

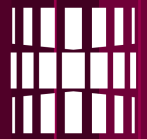
Listing 3.4: Class template for OOP style VHDL

```
package MAC is
    type next_t is record
        ...
    end record;

    type self_t is record
        ...
        nexts: next_t;
    end record;

    -- function prototypes
end package;

package body MAC is
    procedure reset(self: inout self_t) is
        ...
    procedure update_registers(self: inout self_t) is
        ...
    procedure main(self:inout self_t) is
        ...
    -- other user defined functions
end package body;
```

Previous works

„OO-VHDL. Object-oriented extensions to VHDL” by S. Swamy and A. Molin and B. Covnot, 1995

„Object Oriented Extensions to VHDL” by Benzakki, Judith and Djafri, Bachir, 1997



There are verification through simulation and different formal verification methods.
Where is positioned Pyha in respect of those verification methods?

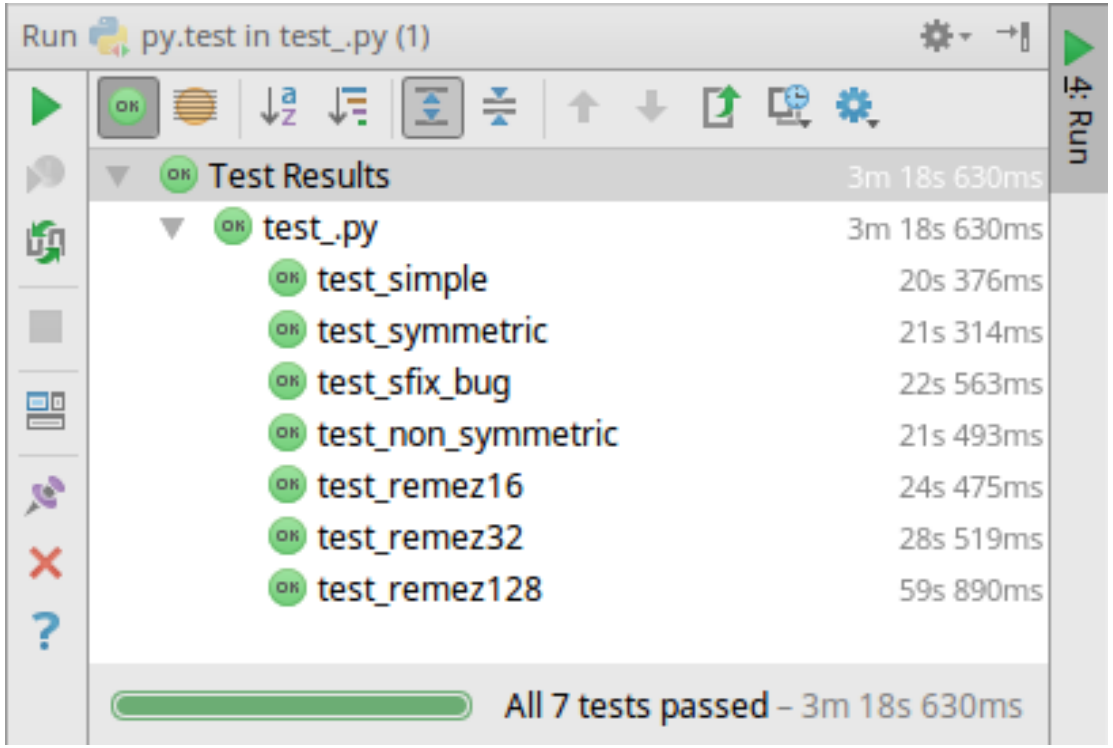
Verification example

```
class FIR(HW):
    """ FIR filter, taps will be normalized to sum 1 """
    def __init__(self, taps):
        self.taps = rescale_taps(taps)

        # registers
        self.acc = [Sfix(left=1, round_style=fixed_truncate, overflow_style=fixed_wrap)] * len(self.taps)
        self.out = Sfix(0, 0, -17, round_style=fixed_truncate)

        # constants
        self.taps_fix_reversed = Const([Sfix(x, 0, -17) for x in reversed(self.taps)])
        self._delay = 2

    def main(self, x):
        """
        Transposed form FIR implementation
        """
        self.acc[0] = x * self.taps_fix_reversed[0]
        for i in range(1, len(self.taps_fix_reversed)):
            self.acc[i] = self.acc[i - 1] + x * self.taps_fix_reversed[i]

        self.out = self.acc[-1]
        return self.out

    def model_main(self, x):
        return signal.lfilter(self.taps, [1.0], x)
```

Verification example

```
def test_simple():
    taps = [0.01, 0.02]
    dut = FIR(taps)
    inp = [0.1, 0.2, 0.3, 0.4]

    assert_simulate(dut, inp)


def test_symmetric():
    taps = [0.01, 0.02, 0.03, 0.04, 0.03, 0.02, 0.01]
    dut = FIR(taps)
    inp = np.random.uniform(-1, 1, 64)

    assert_simulate(dut, inp)
```

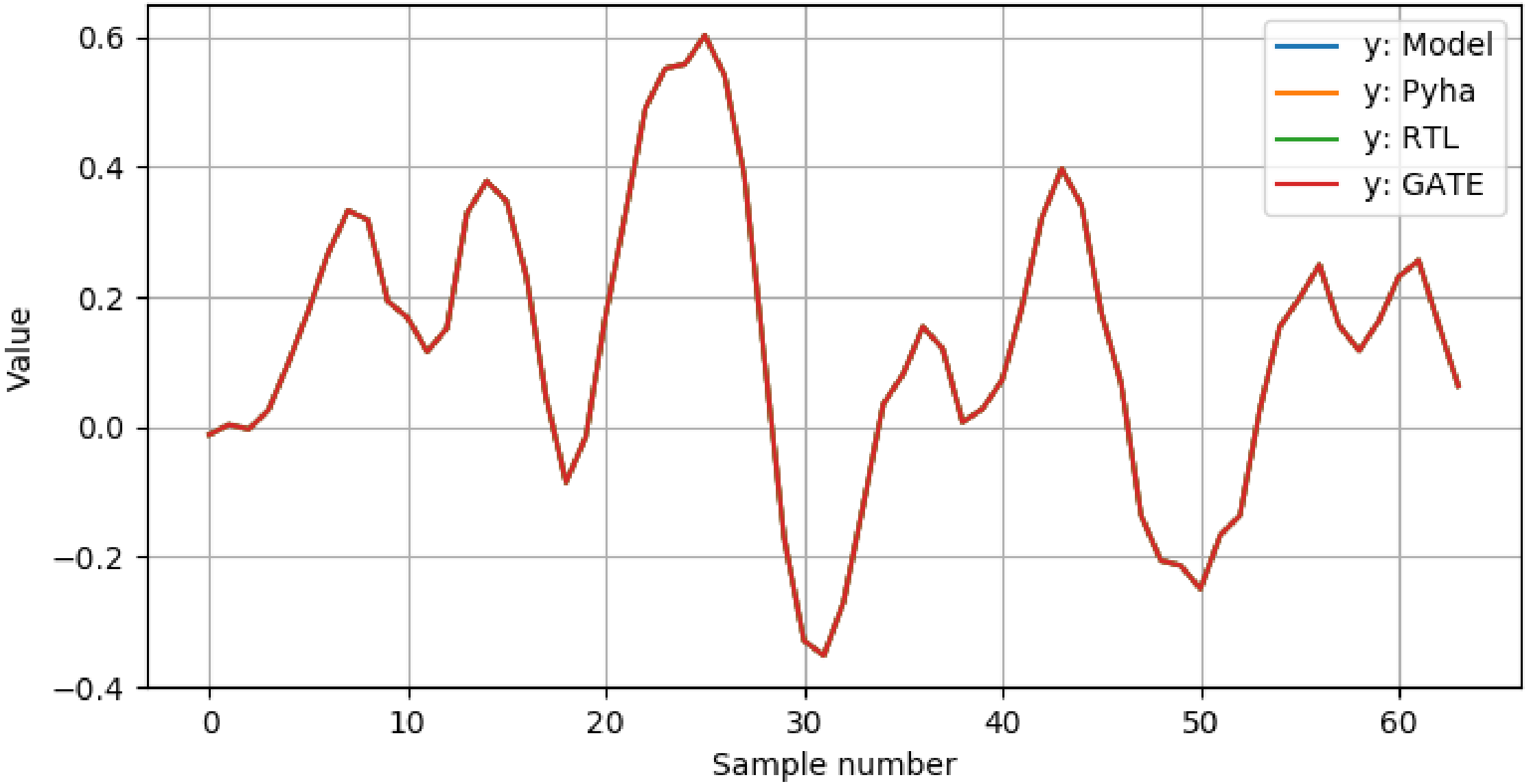
What is ‘boilerplate code’?

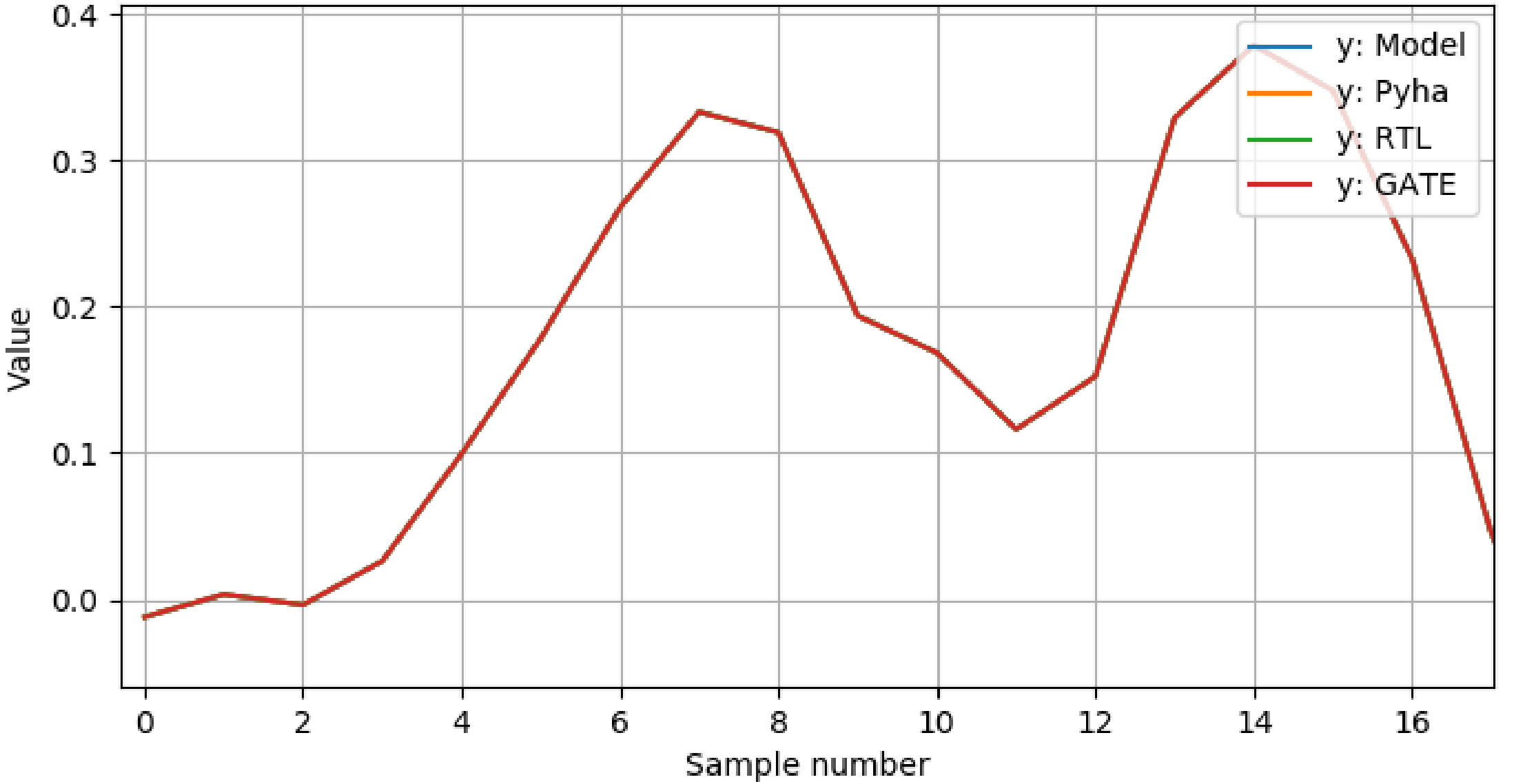


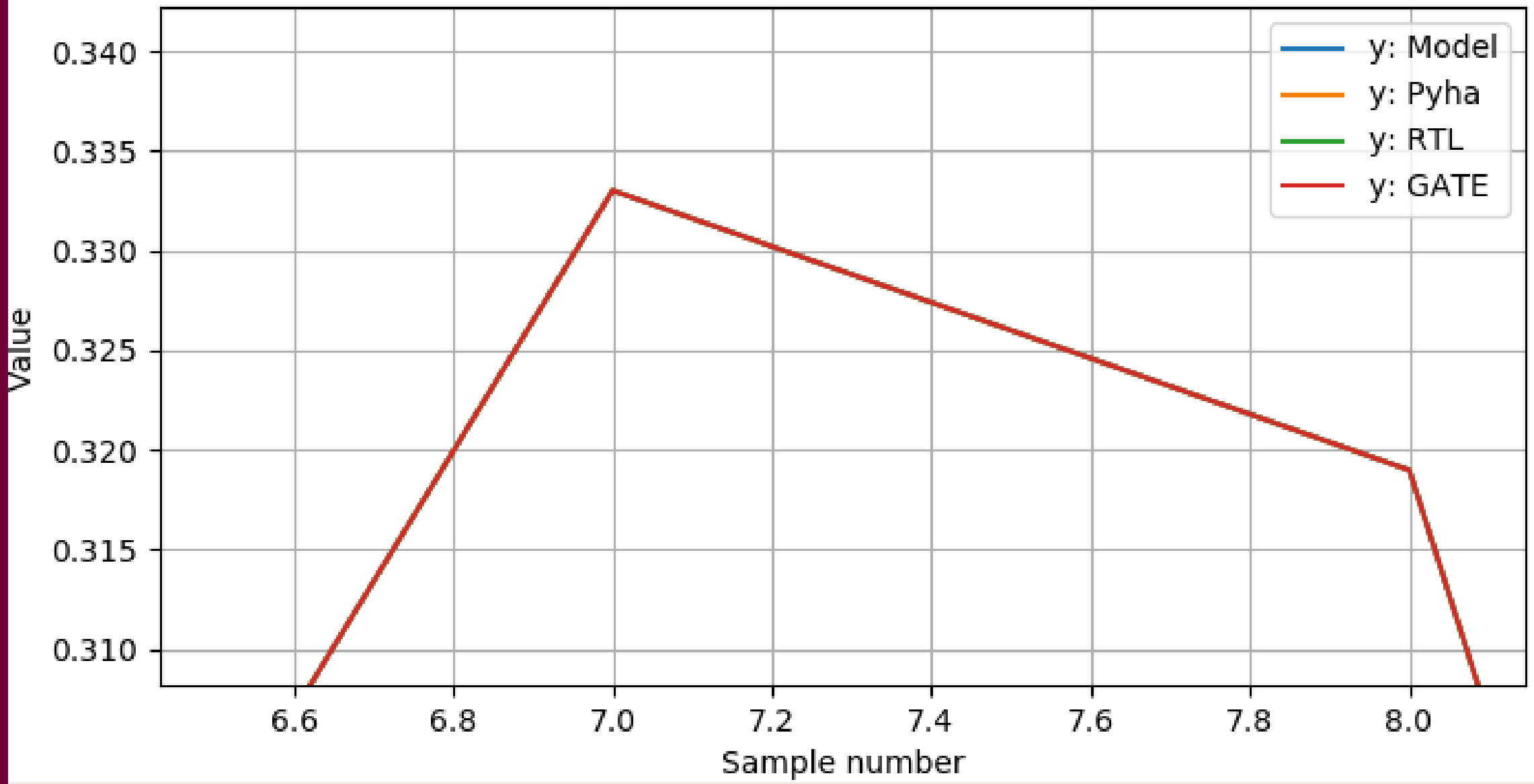
Simulation modes:

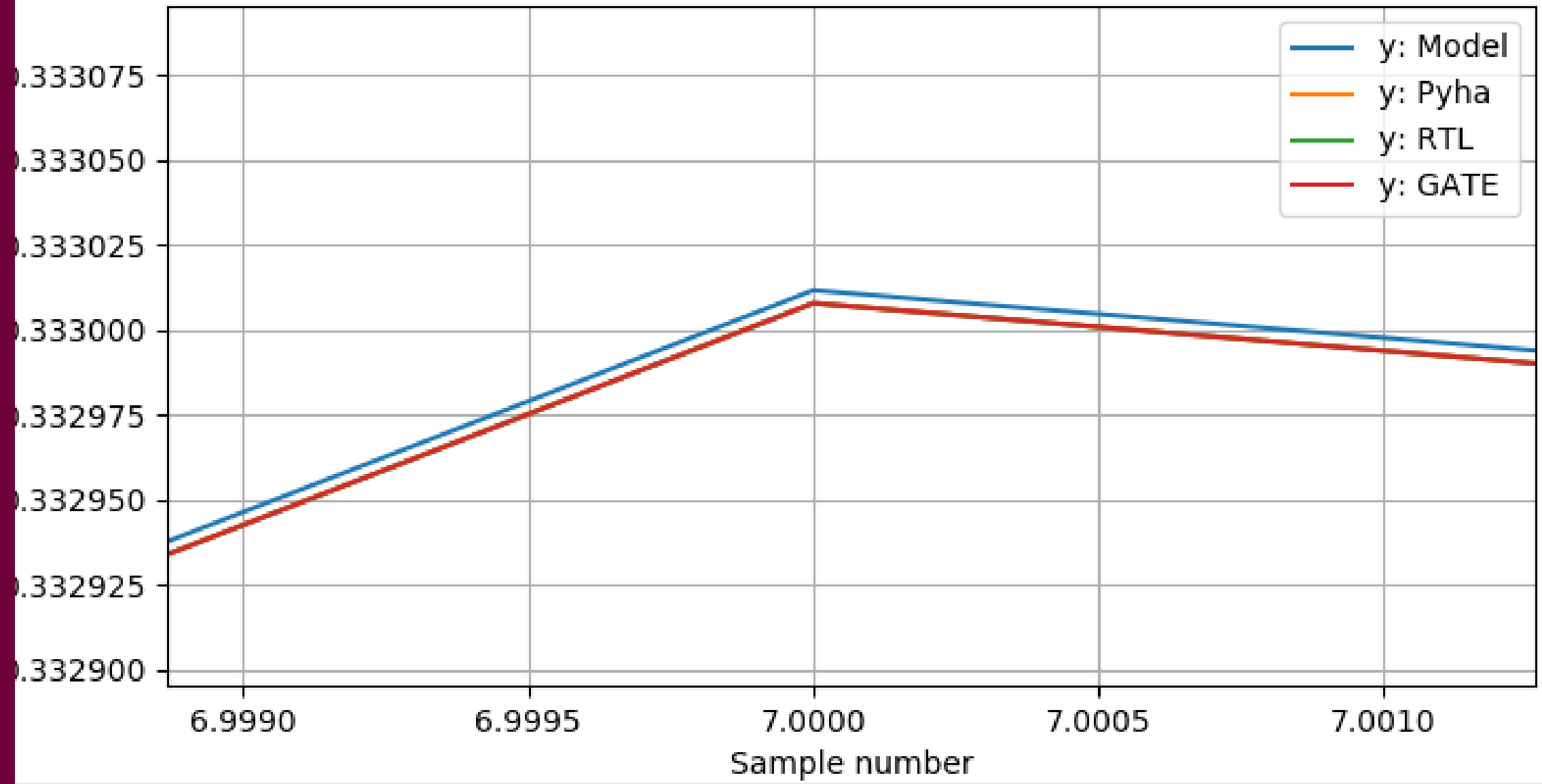
- Model
- Pyha
- RTL
- GATE

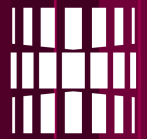
In Figure 2.4. (Pg. 9), Figure 2.9 (Pg. 15) are four elements in Legend, but only two values are visible.


y: Model
y: Pyha
y: RTL
y: GATE
Value
Sample number
0.6
0.4
0.2
0.0
−0.2
−0.4
0
10
20
30
40
50
60


0.4
0.3
0.2
0.1
0.0
Value
0
2
4
6
8
10
12
14
16
Sample number
y: Model
y: Pyha
y: RTL
y: GATE


y: Model
y: Pyha
y: RTL
y: GATE
0.340
0.335
0.330
0.325
0.320
0.315
0.310
Value
6.6
6.8
7.0
7.2
7.4
7.6
7.8
8.0
Sample number


y: Model
y: Pyha
y: RTL
y: GATE
.333075
.333050
.333025
.333000
.332975
.332950
.332925
.332900
6.9990
6.9995
7.0000
7.0005
7.0010
Sample number



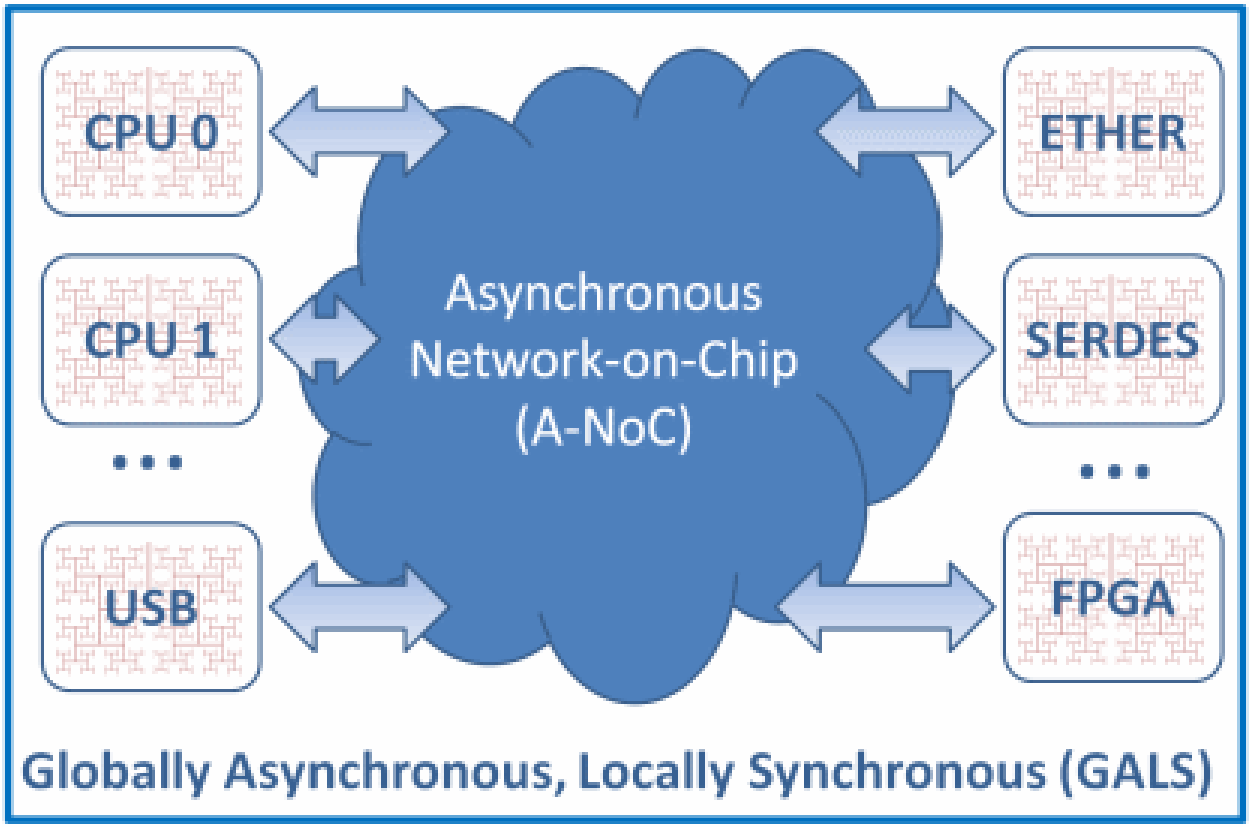
How is Pyha related to MyHDL (which is Python based HDL)?

How is Pyha related to MyHDL (which is Python based HDL)?

- MyHDL is a simulation-oriented language (conversion is very limited)
- Follows the event-driven approach
- Function based designs (not sequential)
- No fixed point type

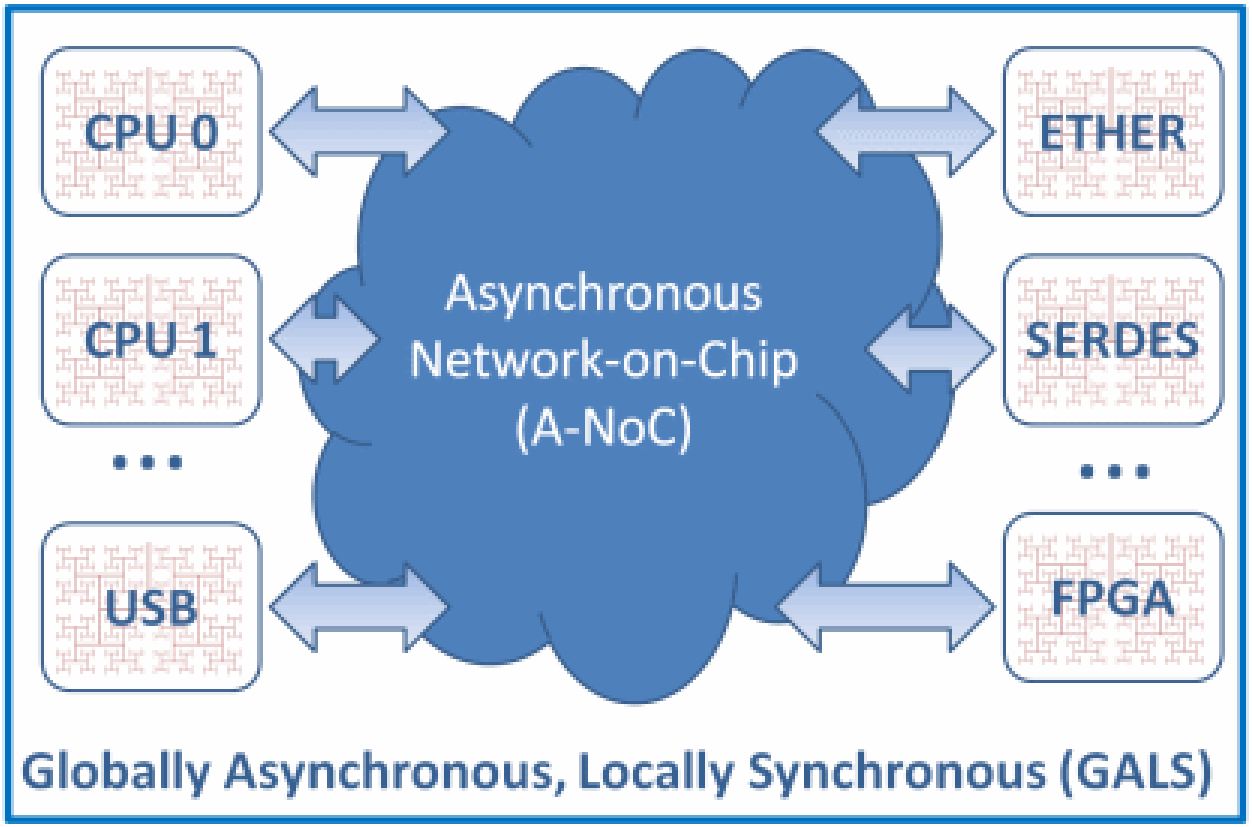
Source: http://www.jandecaluwe.com/blog/its-a-simulation-language.html

How to make Pyha applicable designing GALS (Globally Asynchronous, Locally Synchronous) systems?

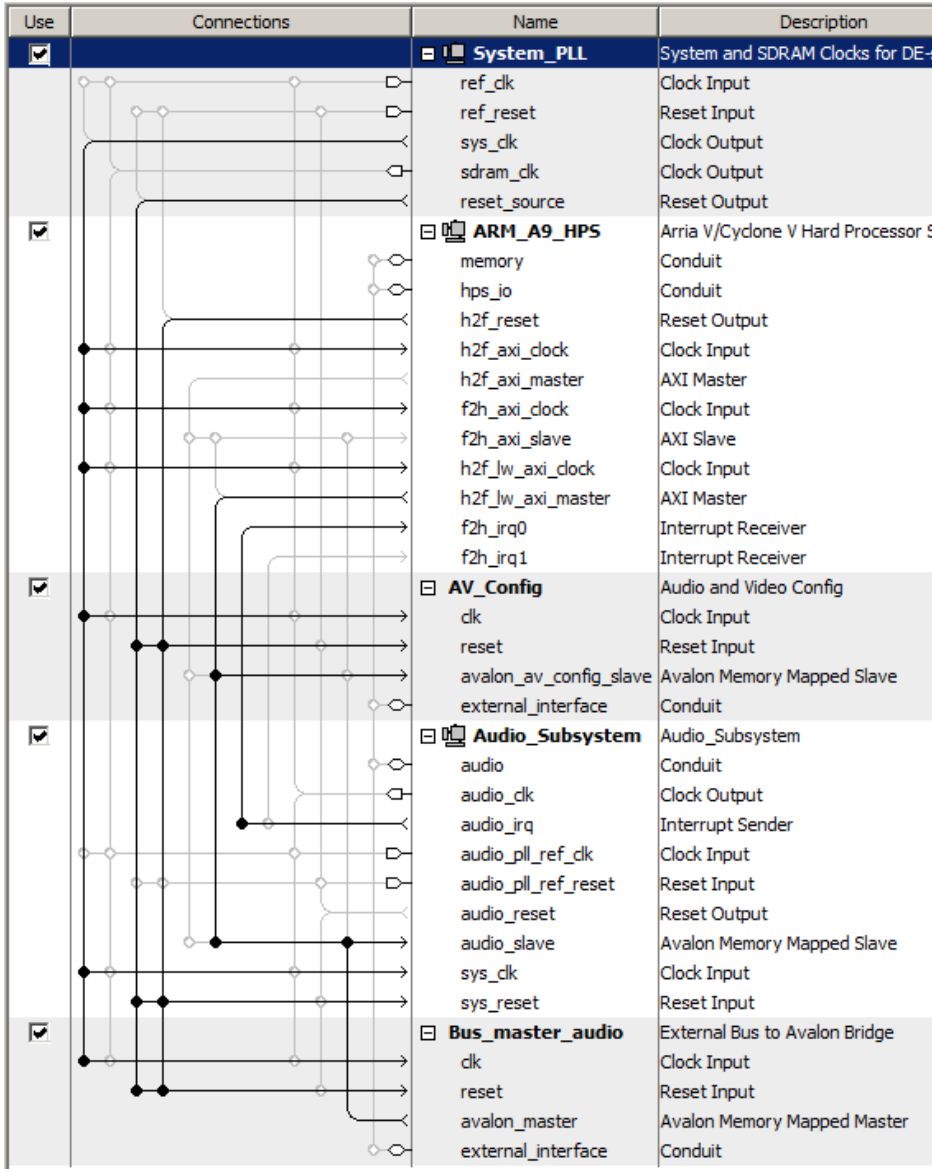


Source: http://www.eetimes.com/author.asp?doc_id=1320994&page_number=2

How to make Pyha applicable designing GALS (Globally Asynchronous, Locally Synchronous) systems?



Source: http://www.eetimes.com/author.asp?doc_id=1320994&page_number=2

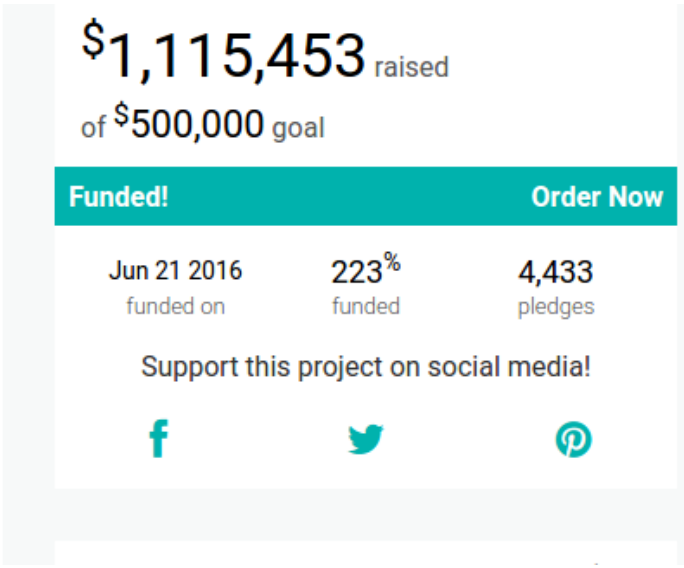


Source: https://hackaday.io/project/19636-audio-dsp-on-de1-soc

The Pyha is aimed for public domain – where can it be found?
Is there user guide of Pyha?
What plans has author disseminating the Pyha in HW designers community?

The Pyha is aimed for public domain – where can it be found?
Is there user guide of Pyha?
What plans has author disseminating the Pyha in HW designers community?

Free software: Apache Software License 2.0
Code repository: https://github.com/gasparka/pyha
Documentation: https://pyha.readthedocs.io



Source: https://www.crowdsupply.com/lime-micro/limesdr

TALLINN UNIVERSITY OF TECHNOLOGY

TALLINN UNIVERSITY OF TECHNOLOGY
Ehitajate street 5, 19086 Tallinn, Estonia
Phone +372 620 2002 (Mon-Fri 8.30 AM – 5.00 PM)
ttu@ttu.ee
ttu.ee