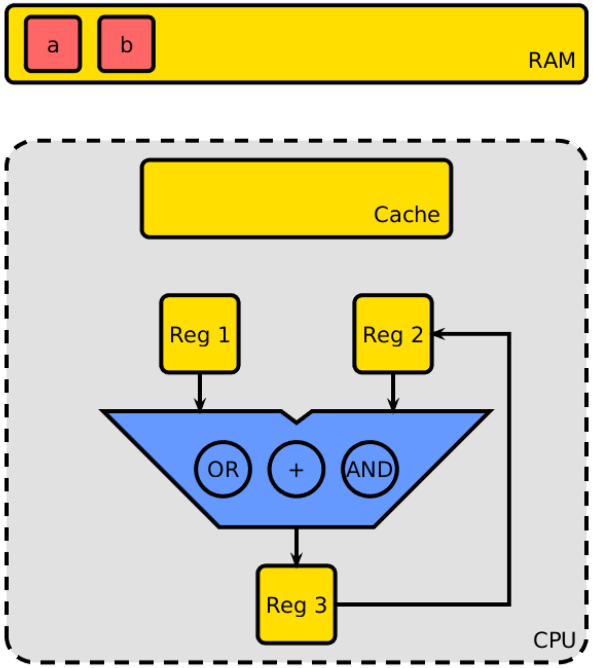
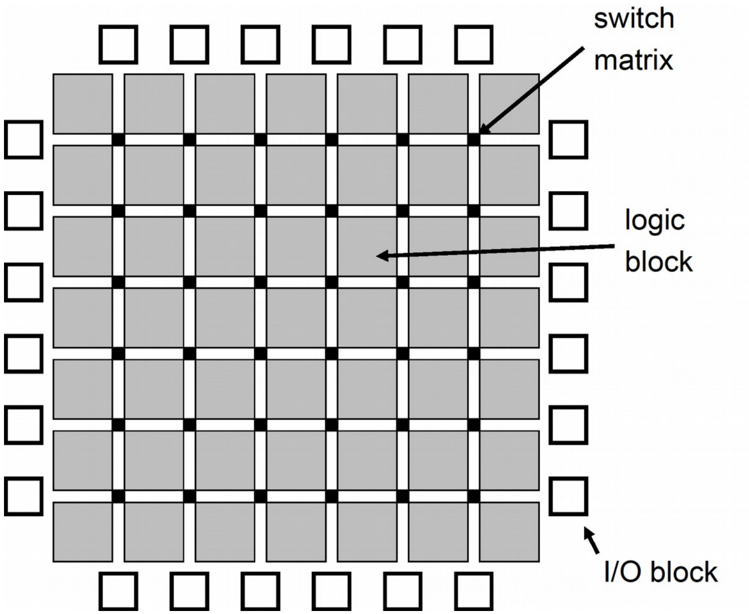
# Introduction to digital circuit design using FPGAs

Bulat Khusainov

TEMPO Summer School on
Hardware Implementation of Embedded Optimisation
17-21 July 2017

# Programming software vs designing hardware



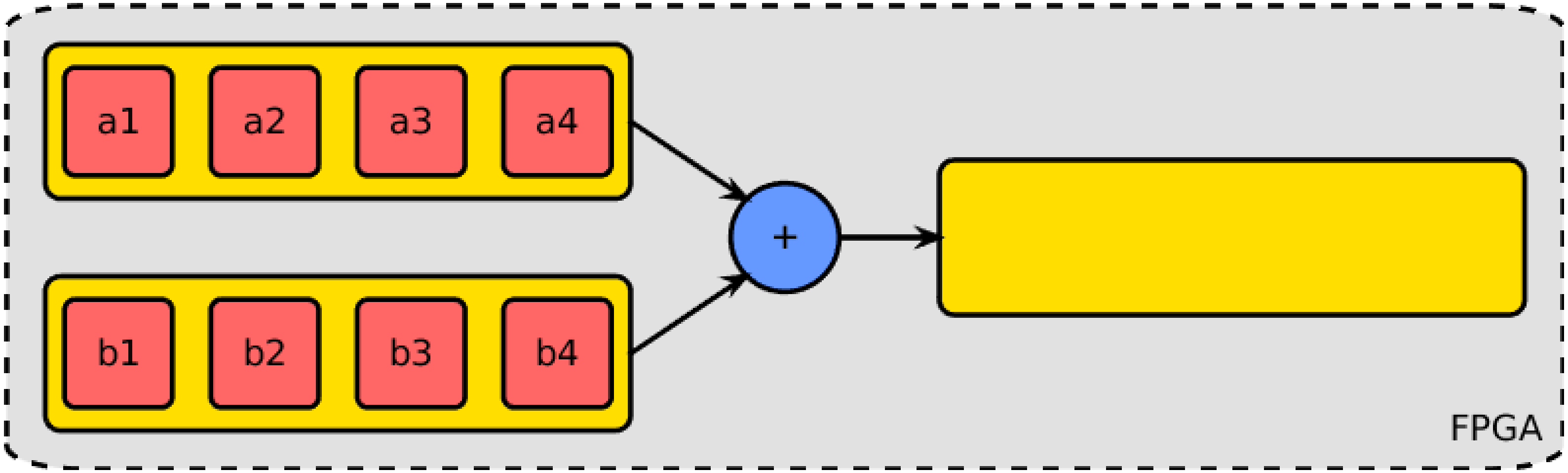


**CPU**:
define a set of operations to be performed

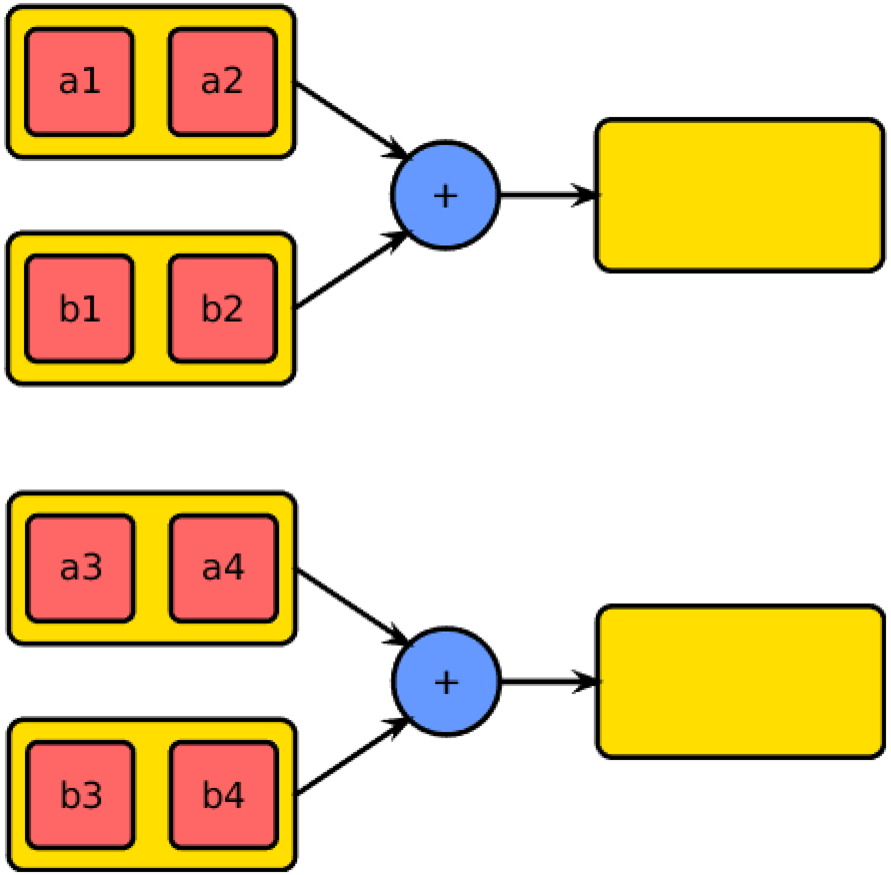
**FPGA**:
describe the circuit i.e. connections between logic blocks

For more information: extended tutorial video.

# FPGA: multiple designs

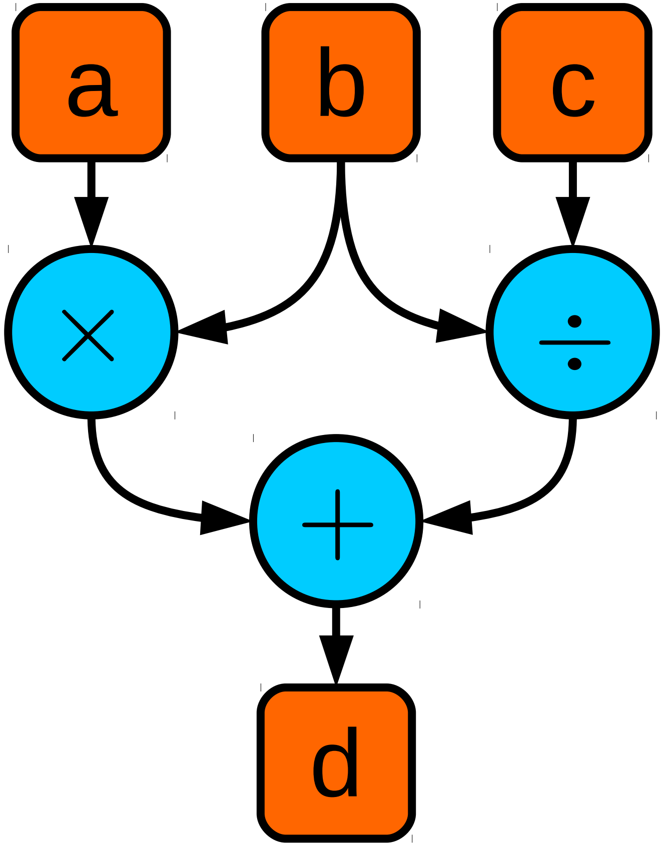


# FPGA design parameters



$$d = (a * b) + (b/c)$$

- **load a**
- **load b**
- *mult*
- **store tmp1**
- **load b**
- **load c**
- *div*
- **store tmp2**
- **load tmp1**
- **load tmp2**
- *add*
- **store d**



# Physics behind computations

Figures sources:

1. http://s5.listing.aystatic.by/c200x200/248/852/5018/5018852248_0.jpg
2. https://regnum.ru/uploads/pictures/news/2016/05/28/regnum_picture_1464384304271457_normal.JPG

# Other computers

- Domino computer: https://www.youtube.com/watch?v=OpLU__bhu2w&t=2s

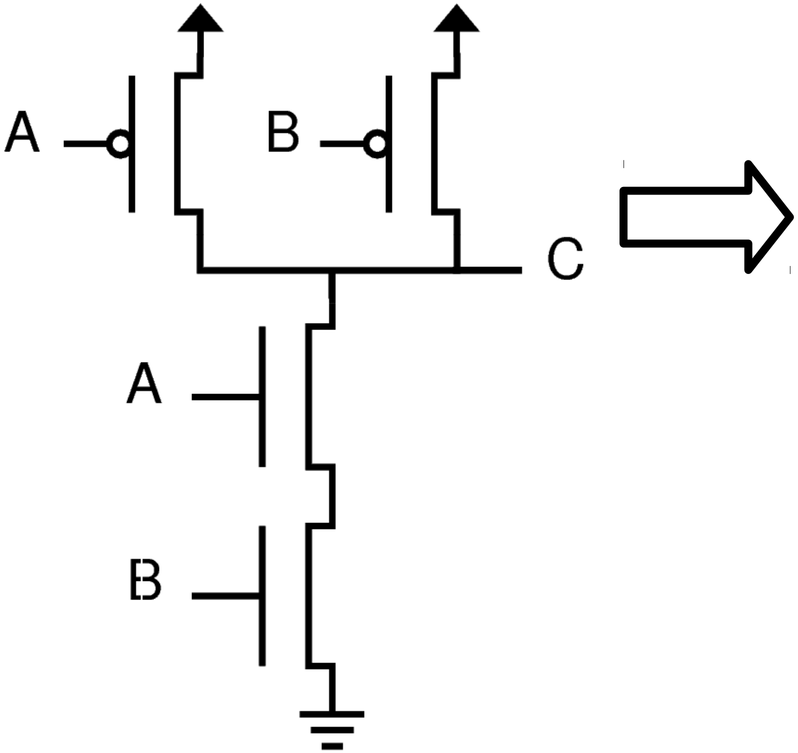
- Relay computer: https://www.youtube.com/watch?v=NXeBR-lbnjI

- Water droplet computer: http://news.stanford.edu/2015/06/08/computer-water-drops-060815/

- Quantum computer: https://www.youtube.com/watch?v=g_IaVepNDT4

# Physics behind computations

C <= A nand B



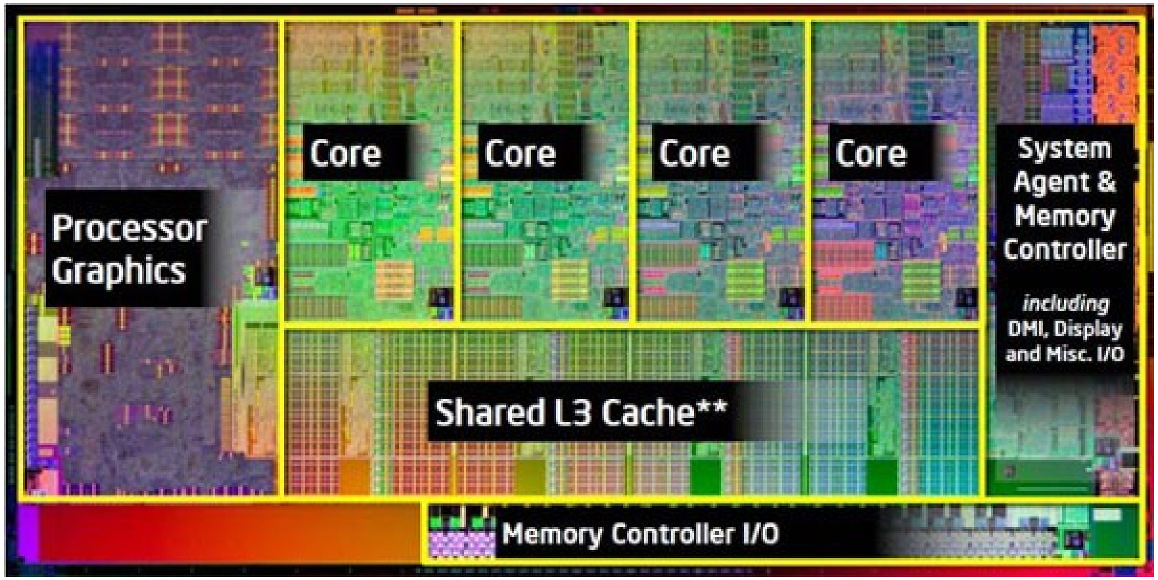
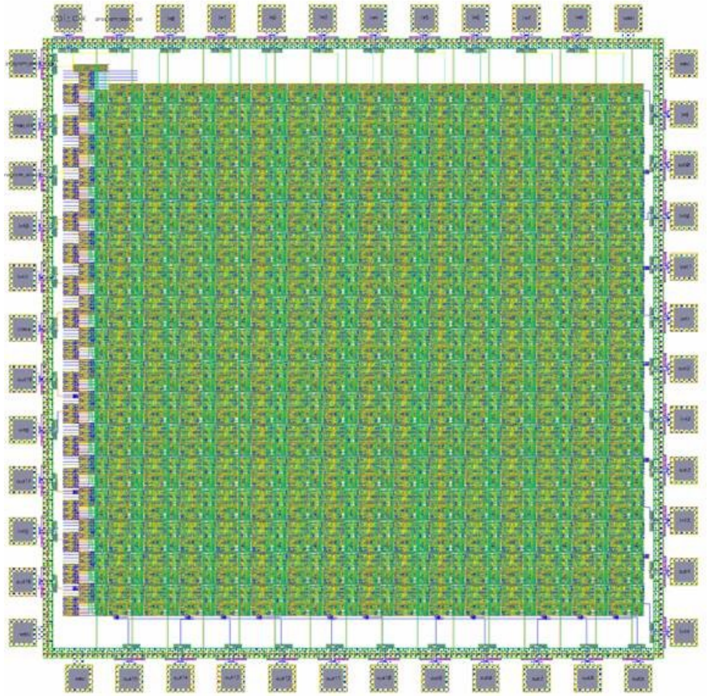


Figure source: http://aphnetworks.com/review/intel_core_i5_2500k/cpu_diagram.jpg

# FPGA vs.CPU



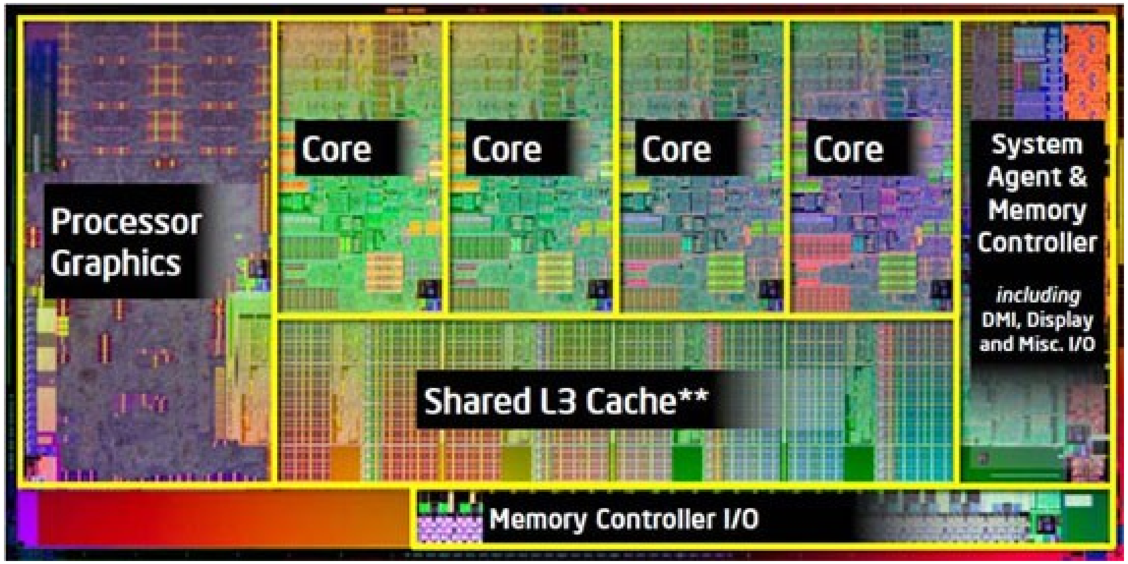
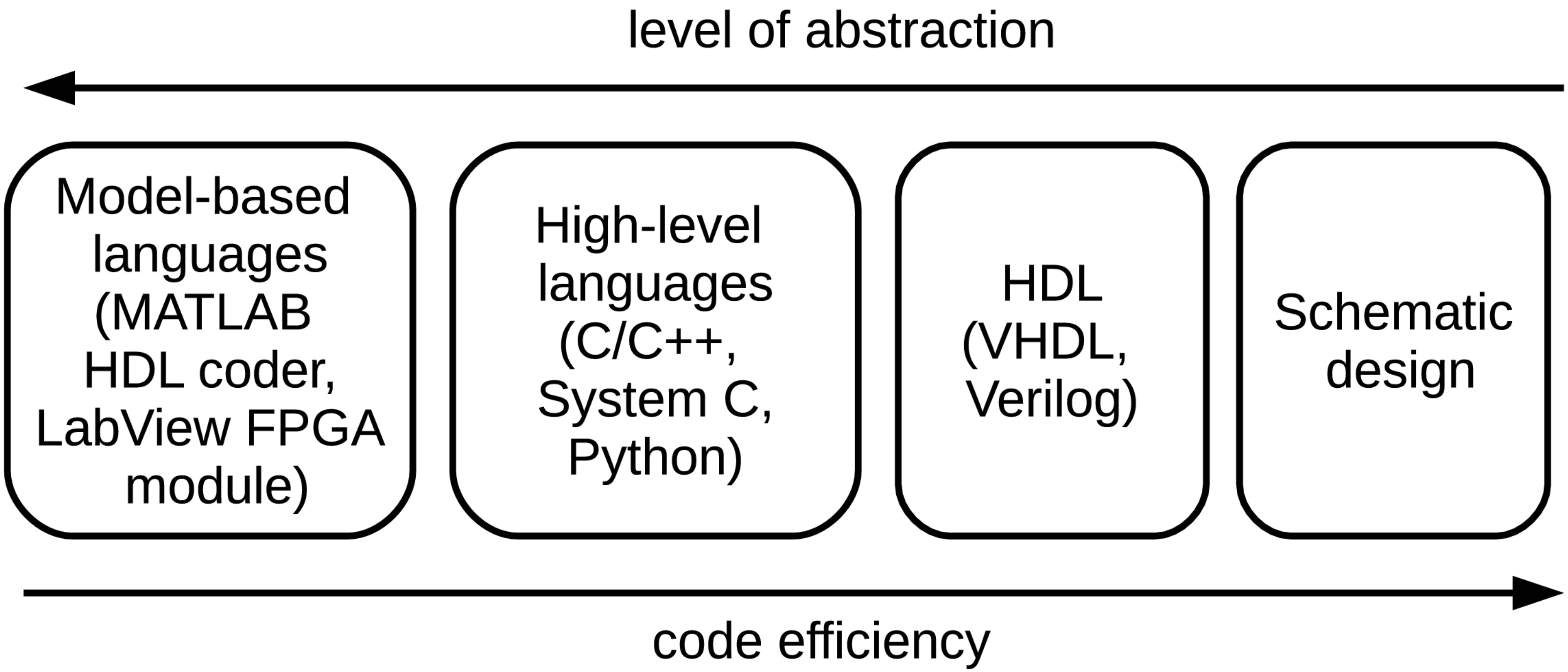


Figure source:
http://www.eecg.toronto.edu/~roman/teaching/1388/2004/finalProj/2004_ECE1388_FP_www/SRAM-Based_FPGA/Final%20Report_files/image003.jpg

# FPGA configuration languages



level of abstraction

Model-based languages (MATLAB HDL coder, LabView FPGA module)

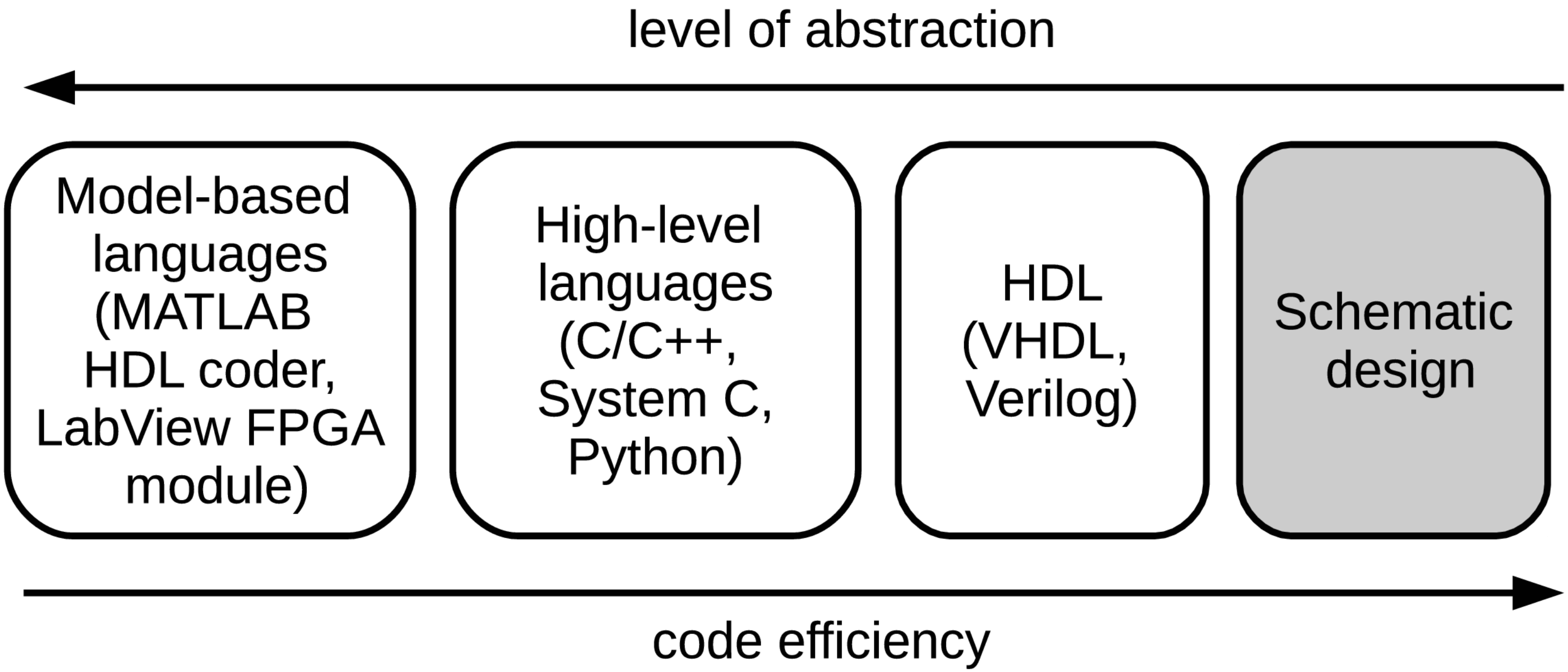
High-level languages (C/C++, System C, Python)

HDL (VHDL, Verilog)

Schematic design

code efficiency

# FPGA configuration languages



level of abstraction

Model-based languages (MATLAB HDL coder, LabView FPGA module)

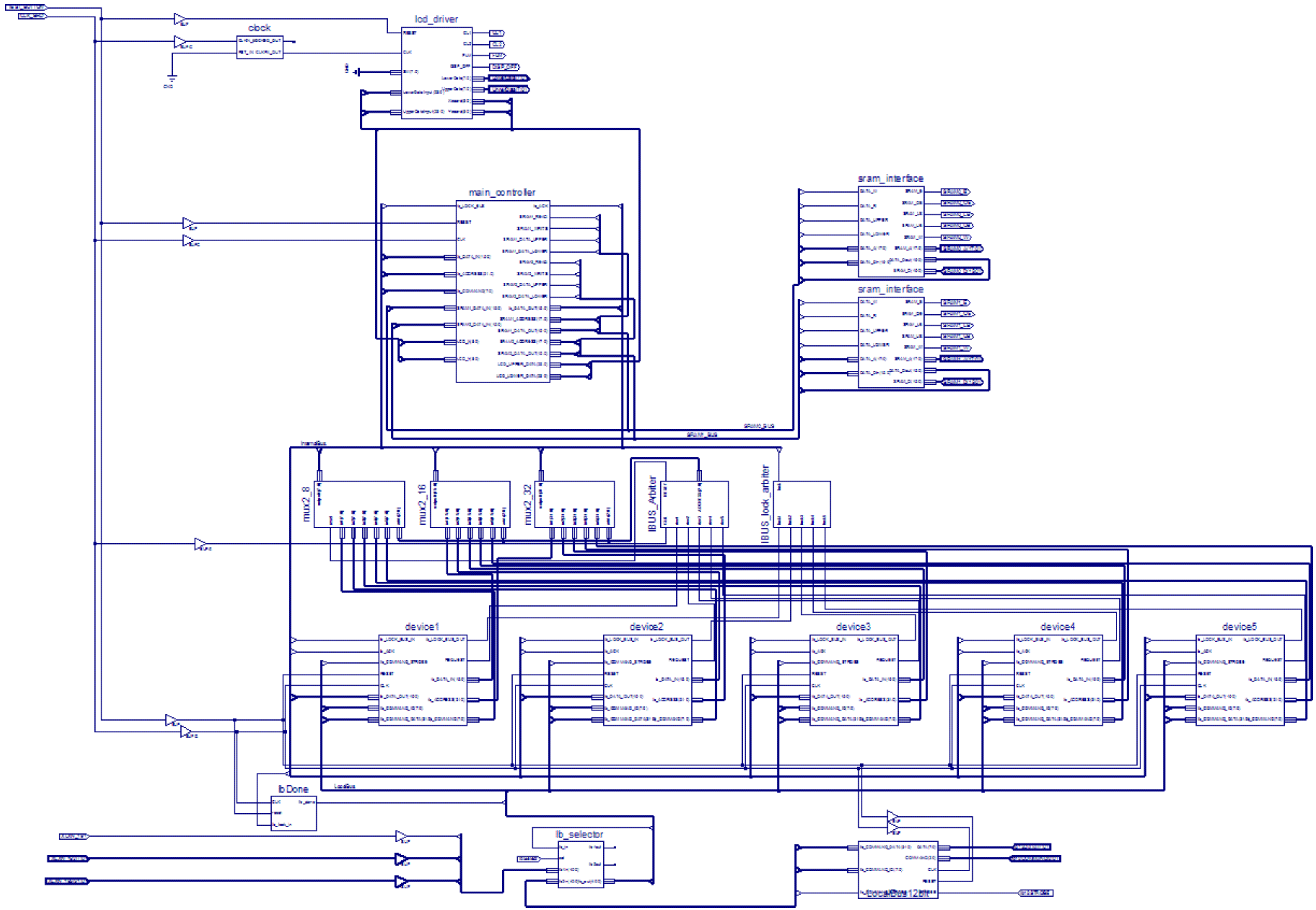
High-level languages (C/C++, System C, Python)

HDL (VHDL, Verilog)

Schematic design

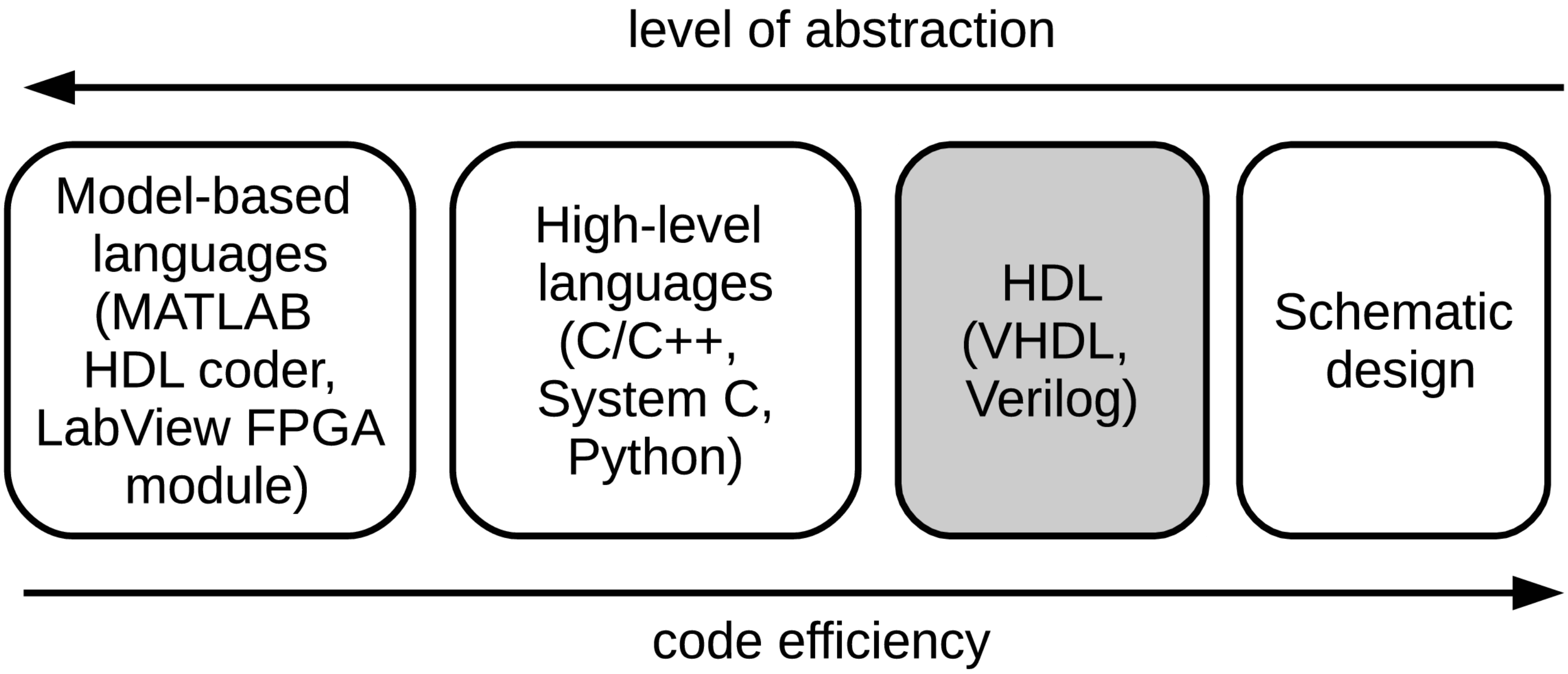
code efficiency

# Schematic design



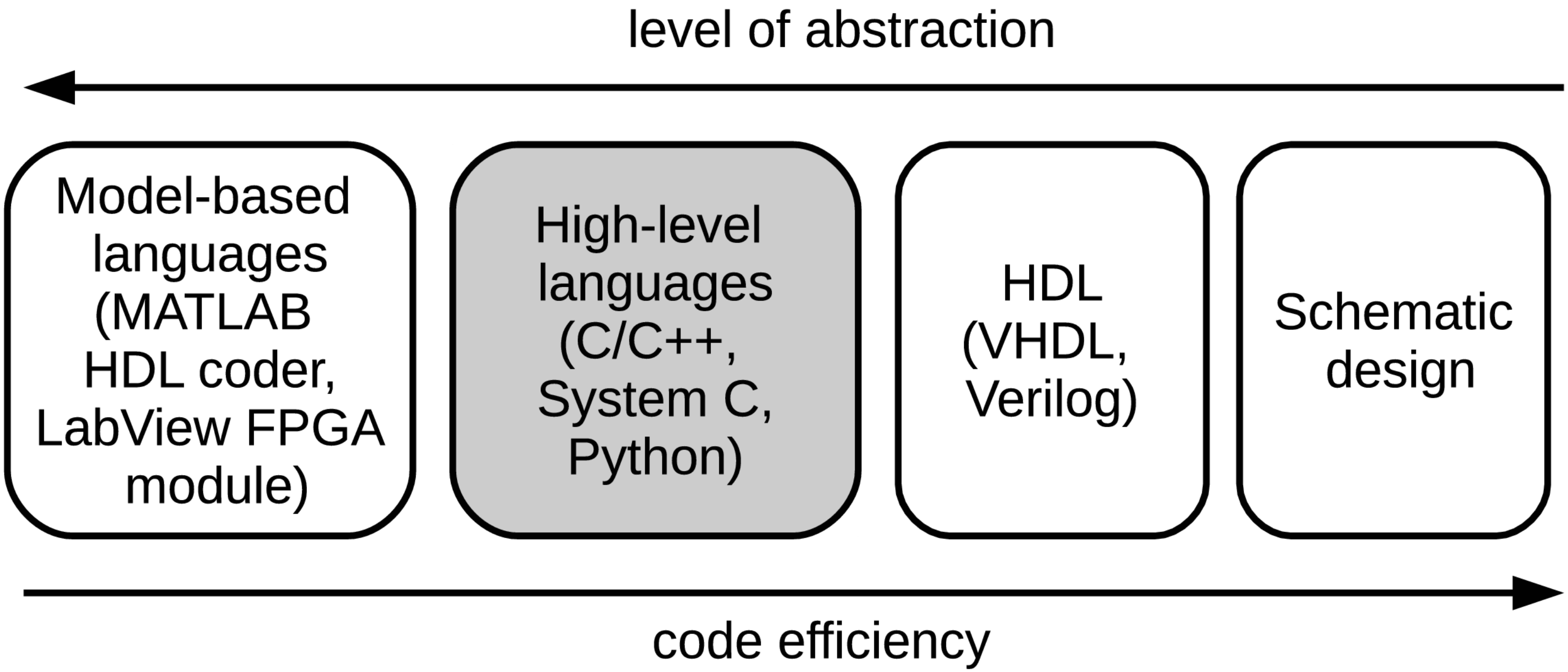
For details refer to: http://www.mbr-team.net/electronic/LCDcontroller/

# FPGA configuration languages


level of abstraction
Model-based languages (MATLAB HDL coder, LabView FPGA module)
High-level languages (C/C++, System C, Python)
HDL (VHDL, Verilog)
Schematic design
code efficiency

Circuit design with VHDL will be considered in **Labs 1 and 2**

# FPGA configuration languages



level of abstraction

Model-based languages (MATLAB HDL coder, LabView FPGA module)

High-level languages (C/C++, System C, Python)

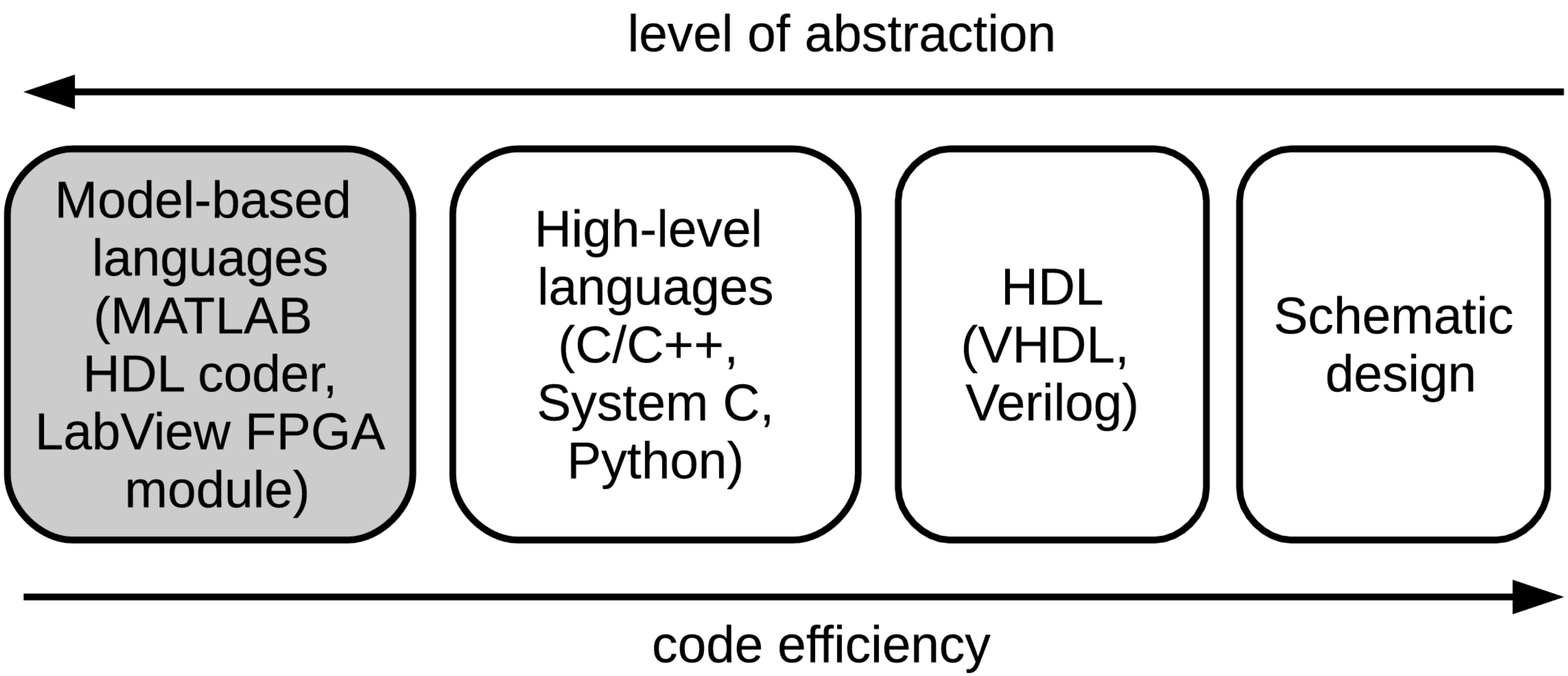
HDL (VHDL, Verilog)

Schematic design

code efficiency

Circuit design with C-based HLS will be considered in **Lab 3**

# FPGA configuration languages



level of abstraction

Model-based languages (MATLAB HDL coder, LabView FPGA module)

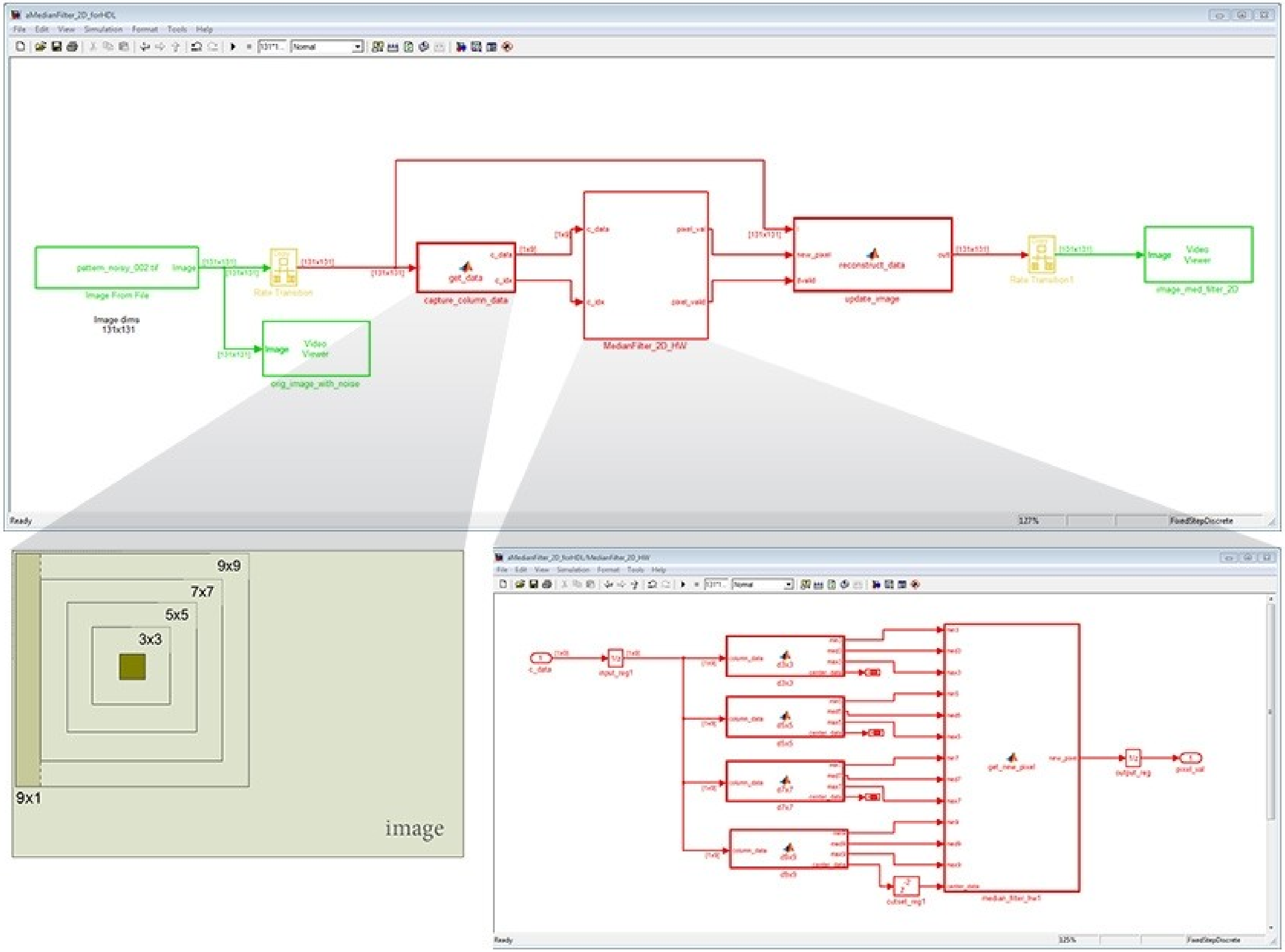
High-level languages (C/C++, System C, Python)

HDL (VHDL, Verilog)

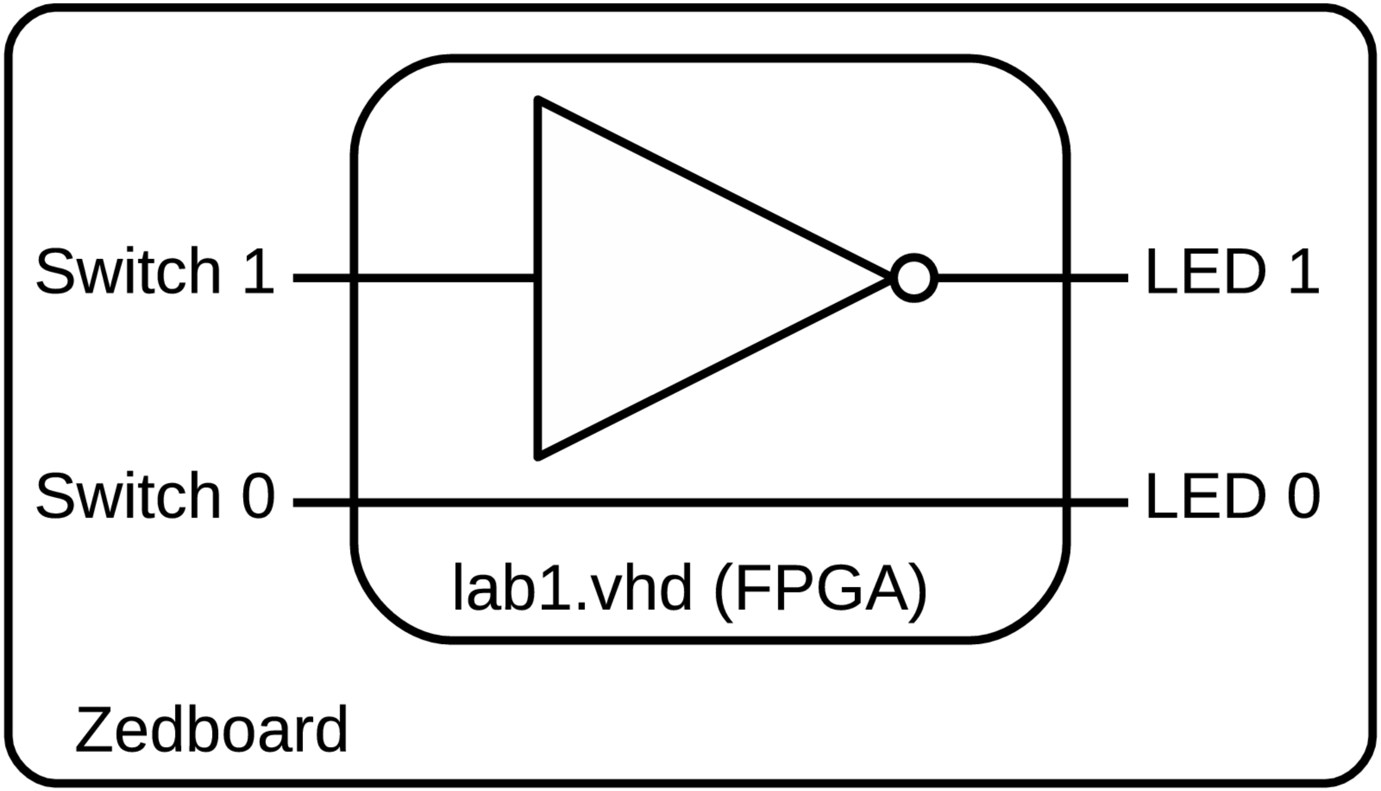
Schematic design

code efficiency

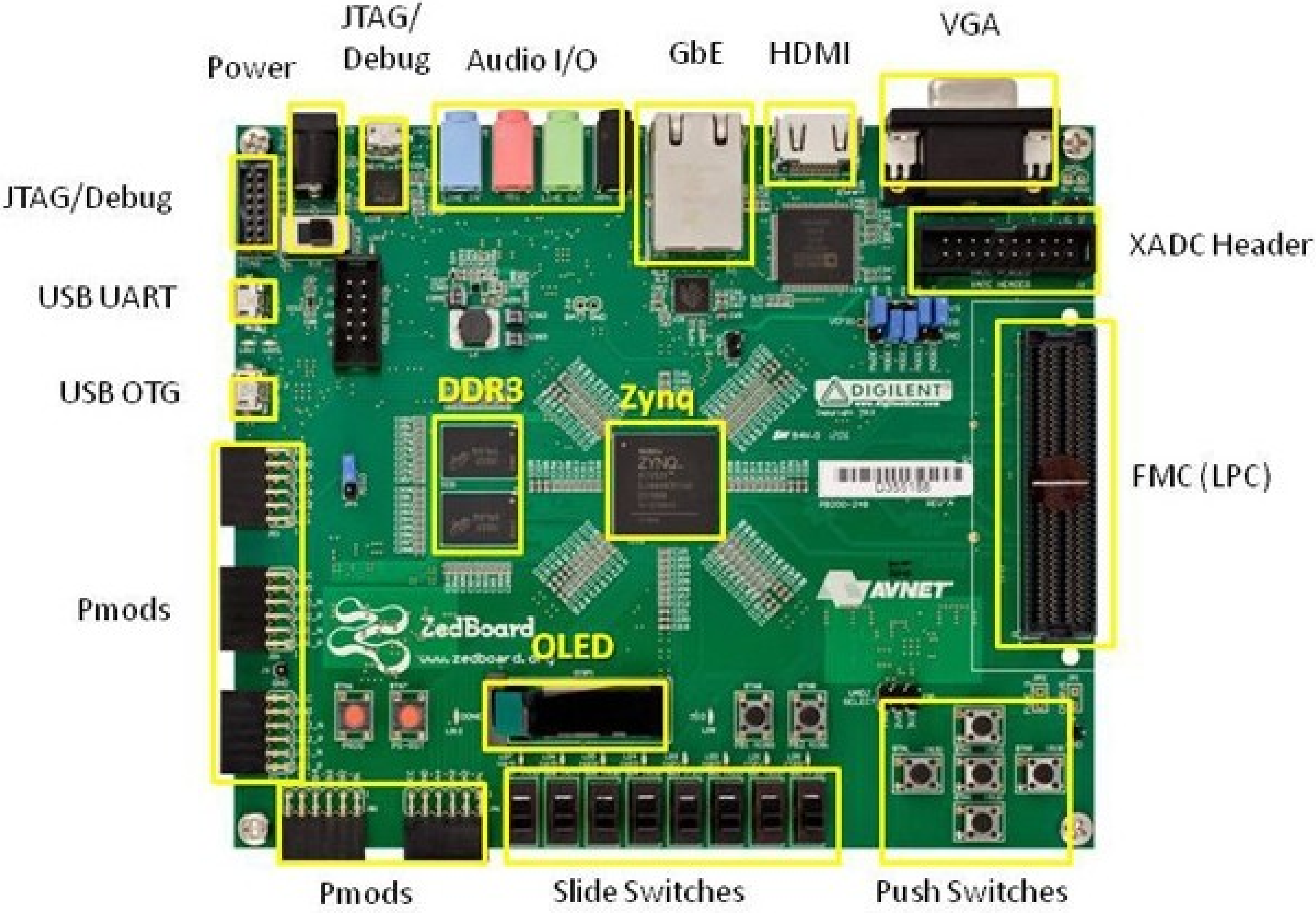
# Model-based design
# (e.g. MATLAB HDL coder)



# Lab1: Introduction to VHDL and Vivado



# Lab1: Introduction to VHDL and Vivado

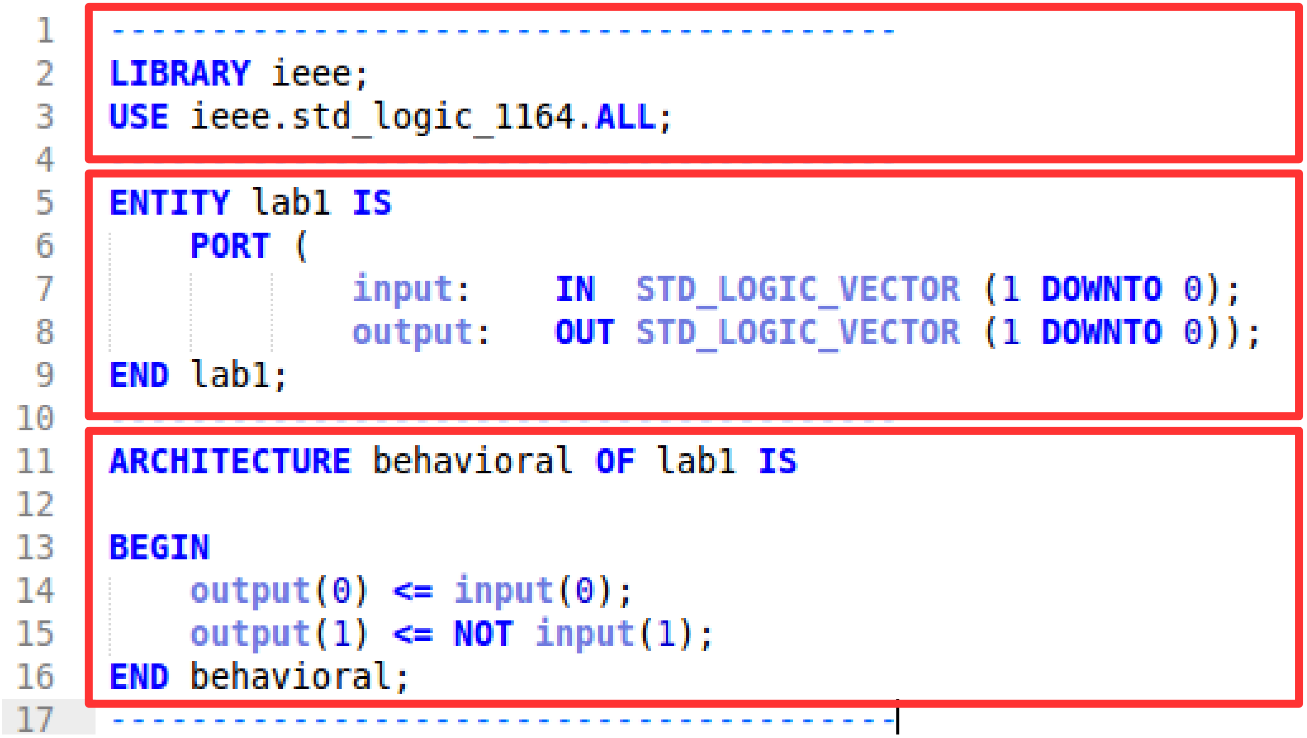


# VHDL code

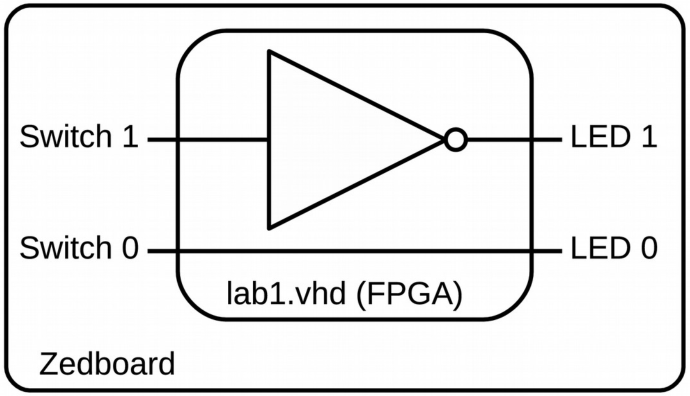
Libraries

Circuit interfaces

Circuit architecture

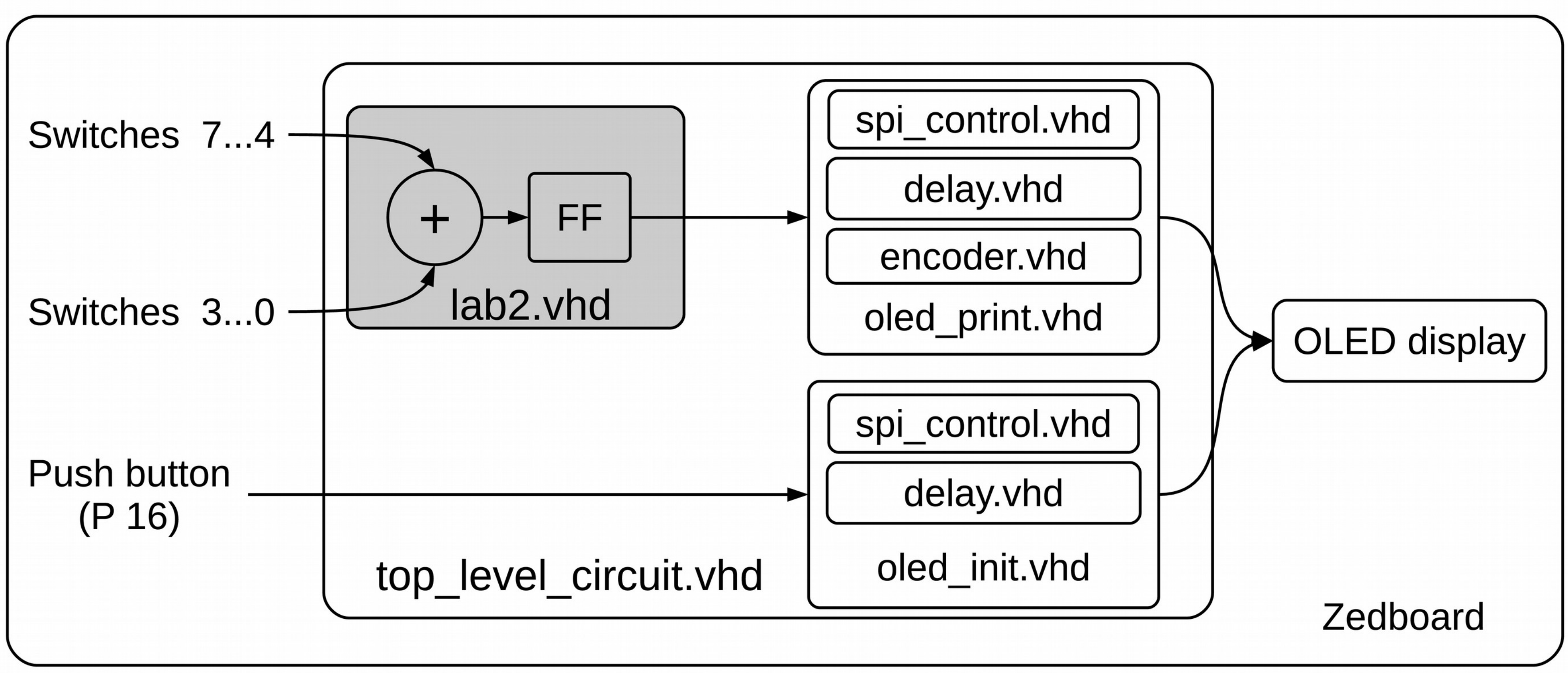


```
---------------------------------------
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY lab1 IS
    PORT (
            input:    IN  STD_LOGIC_VECTOR (1 DOWNTO 0);
            output:   OUT STD_LOGIC_VECTOR (1 DOWNTO 0));
END lab1;
---------------------------------------
ARCHITECTURE behavioral OF lab1 IS

BEGIN
    output(0) <= input(0);
    output(1) <= NOT input(1);
END behavioral;
---------------------------------------
```

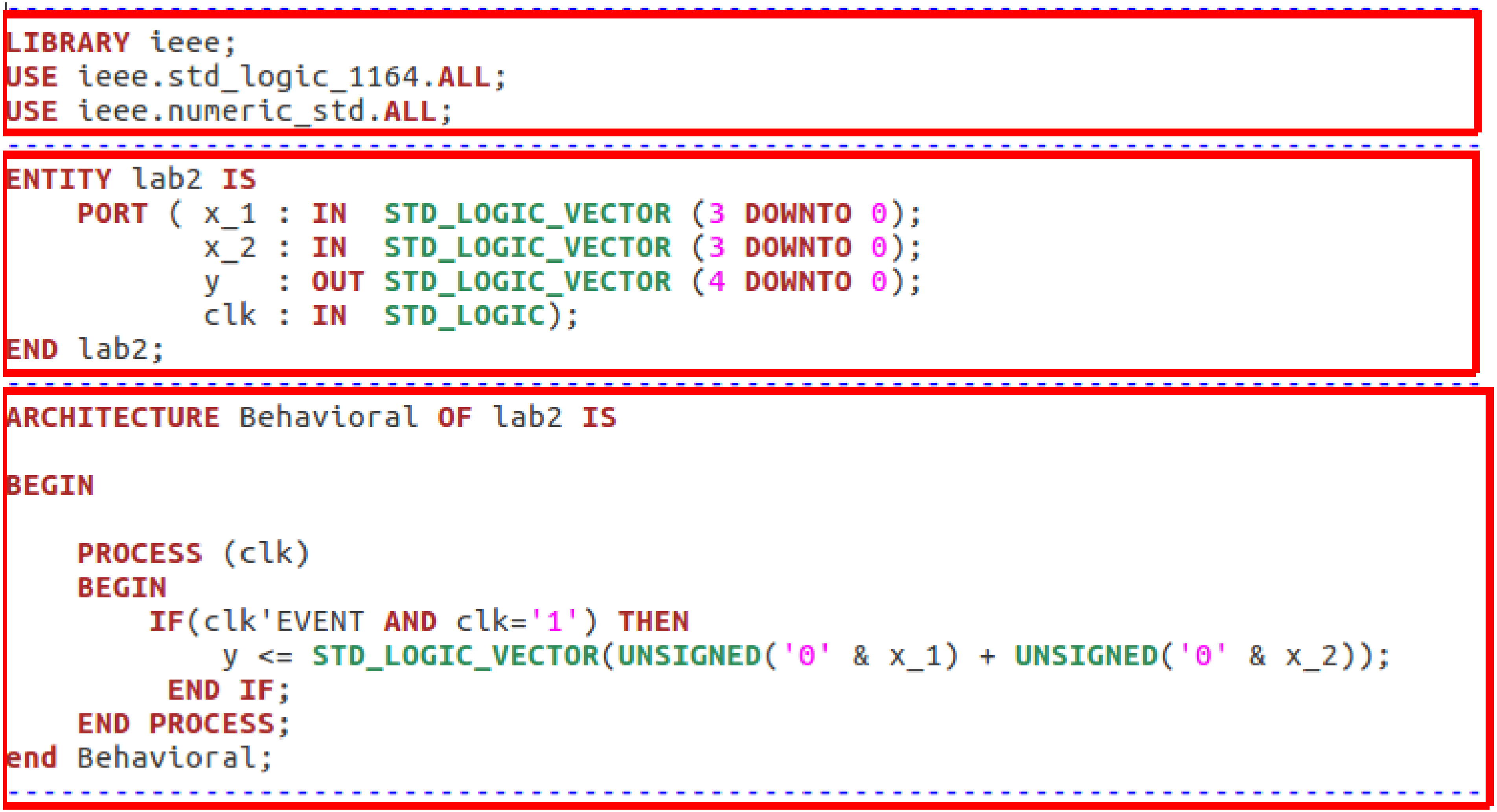


# Lab1: time to practice!

# Lab2: Implementing adder circuit with VHDL

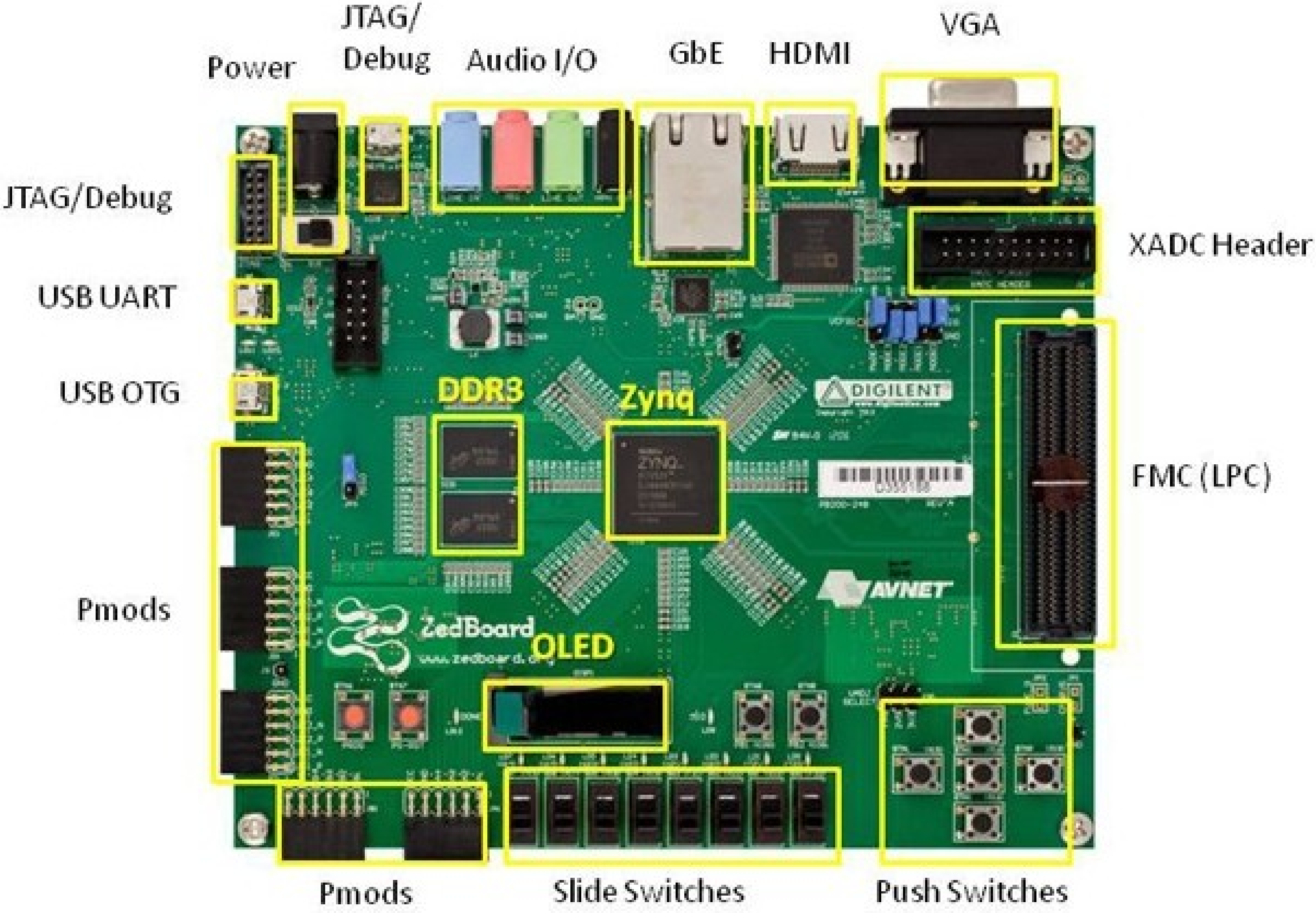


# VHDL code

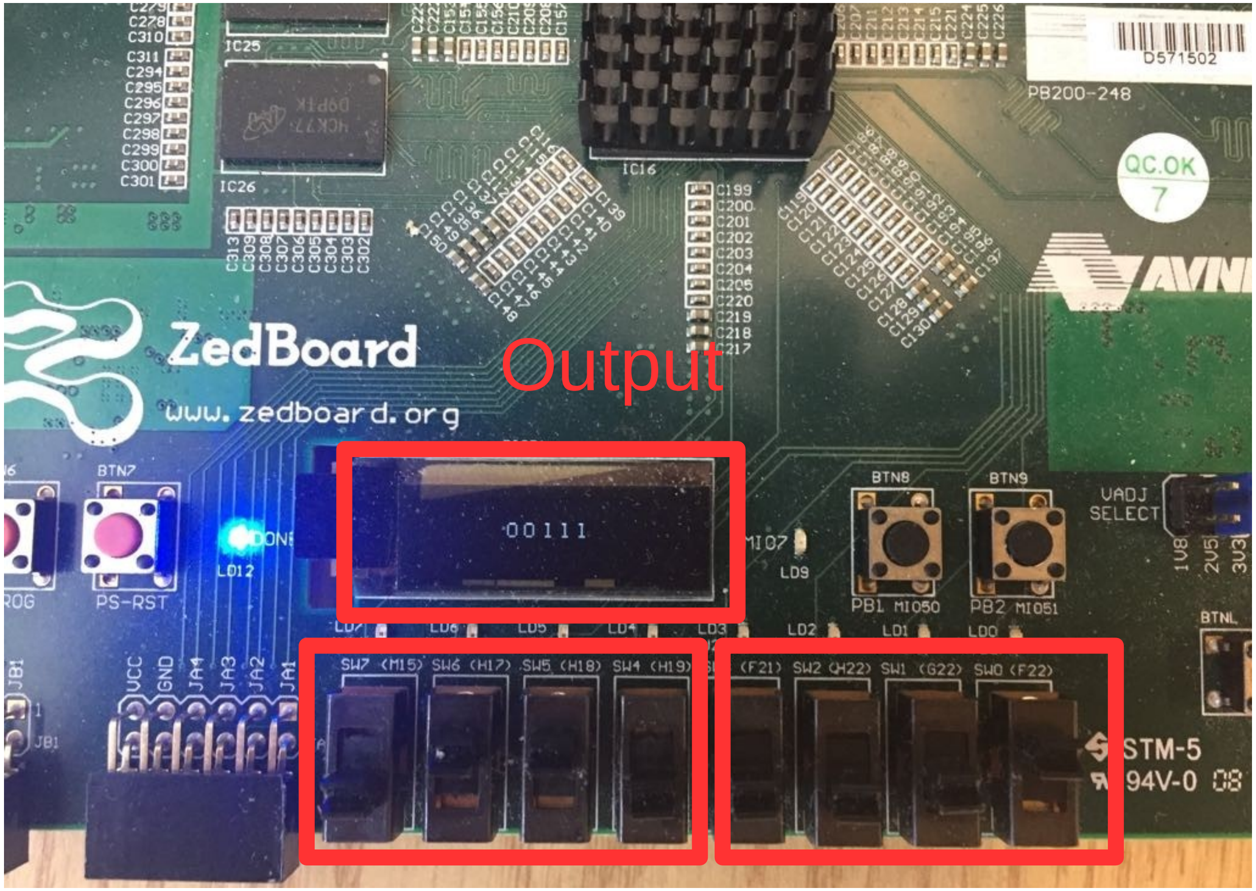


```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.numeric_std.ALL;

ENTITY lab2 IS
    PORT ( x_1 : IN  STD_LOGIC_VECTOR (3 DOWNTO 0);
           x_2 : IN  STD_LOGIC_VECTOR (3 DOWNTO 0);
           y   : OUT STD_LOGIC_VECTOR (4 DOWNTO 0);
           clk : IN  STD_LOGIC);
END lab2;

ARCHITECTURE Behavioral OF lab2 IS

BEGIN

    PROCESS (clk)
    BEGIN
        IF(clk'EVENT AND clk='1') THEN
            y <= STD_LOGIC_VECTOR(UNSIGNED('0' & x_1) + UNSIGNED('0' & x_2));
         END IF;
    END PROCESS;
end Behavioral;
```

# Lab2: Implementing adder circuit with VHDL



# Lab2: Implementing adder circuit with VHDL



Output

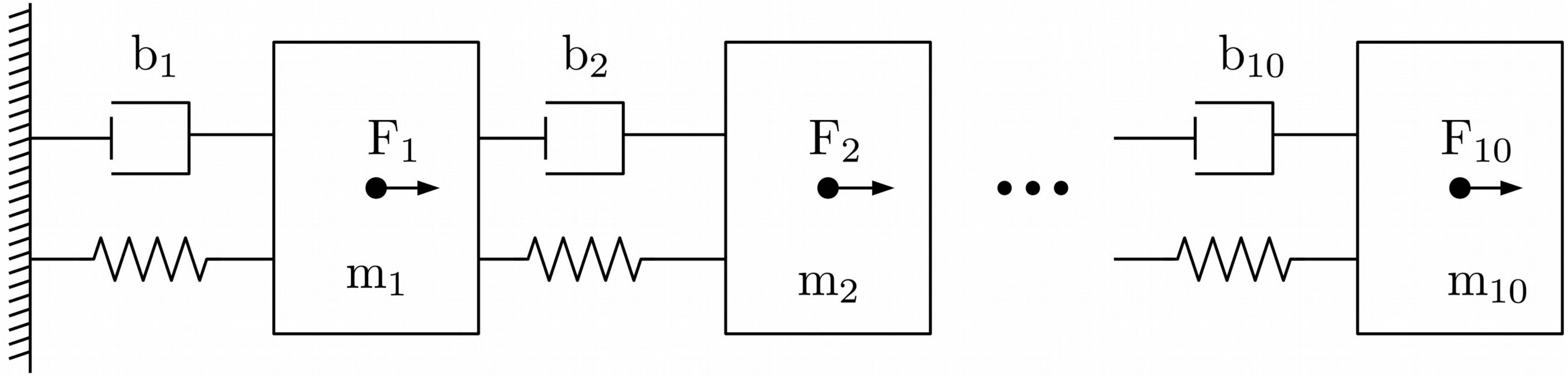
Input 1

Input 2

# Lab2: time to practice!

# Lab 3: Fast gradient algorithm-based predictive control

$$\begin{aligned}
\underset{u_0 \dots u_{N-1}, x_0 \dots x_N}{\text{minimize}} \quad & \sum_{k=0}^{N-1} \left( \frac{1}{2} x_k^T Q_d x_k + \frac{1}{2} u_k^T R_d u_k \right) + \frac{1}{2} x_N^T P_d x_N && (1) \\
\text{subject to} \quad & x_0 = \hat{x} && (2) \\
& x_{k+1} = A_d x_k + B_d u_k, \quad \text{for } k = 0, 1, \dots, N-1 && (3) \\
& u_{min} \leq u_k \leq u_{max}, \quad \text{for } k = 0, 1, \dots, N-1 && (4)
\end{aligned}$$



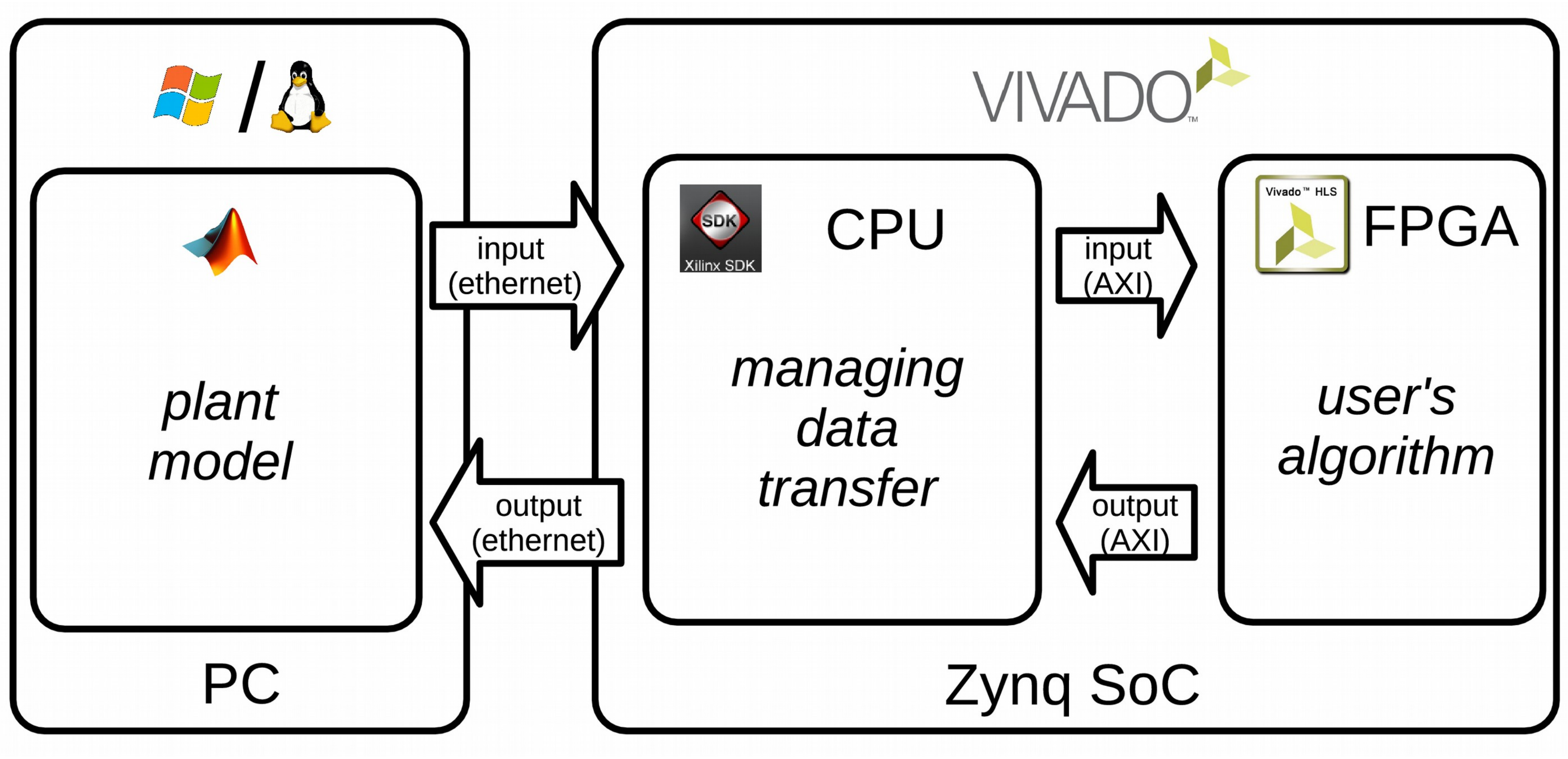
# Lab 3: Fast gradient algorithm-based predictive control

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2}\theta^T H\theta + \theta^T h \\ \text{subject to} \quad & \theta_{min} \leq \theta \leq \theta_{max} \end{aligned}$$

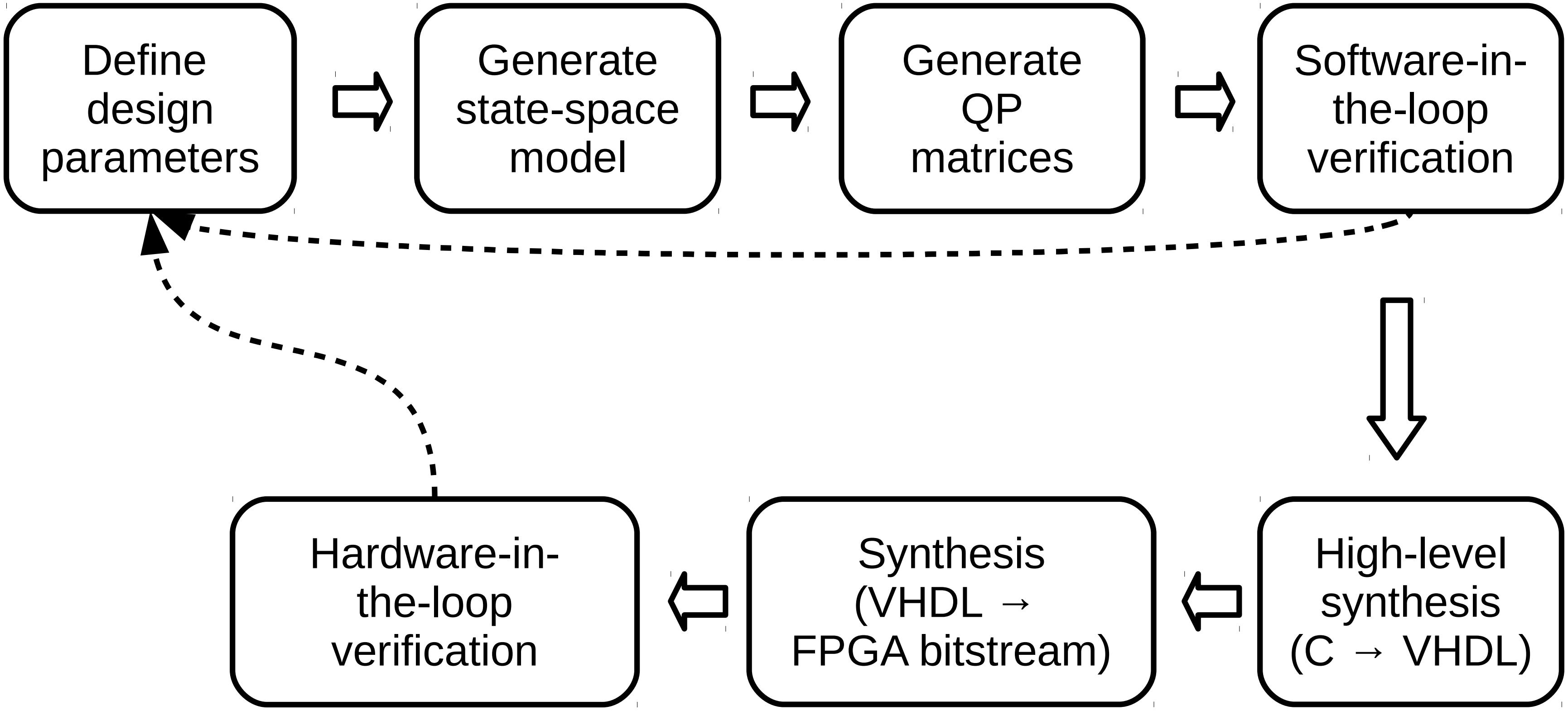
**Algorithm 0.1:** Projected fast gradient algorithm for constrained optimization with constant step size.

```
Initial guess:$\theta_0$
$v_0 = \theta_0$
for $i = 0$ to $N_{FGM}$ do
    $\theta_{i+1} = (I - (1/L)H)\nu_i - (1/L)h$ {anti-gradient step}
    $z_{i+1} = P(\theta_{i+1})$ {projection on the feasible set}
    $v_{i+1} = (1+\beta)z_{i+1} - \beta z_i$ {extra-momentum step}
end for
```

# Hardware verification with Protoip



# Lab3: Design Flow



Define design parameters

Generate state-space model

Generate QP matrices

Software-in-the-loop verification

High-level synthesis (C → VHDL)

Synthesis (VHDL → FPGA bitstream)

Hardware-in-the-loop verification

# Useful links

- Protoip wiki: https://github.com/asuardi/protoip/wiki

- Introduction to Vivado HLS: http://users.ece.utexas.edu/~gerstl/ee382v_f14/soc/vivado_hls/VivadoHLS_Overview.pdf