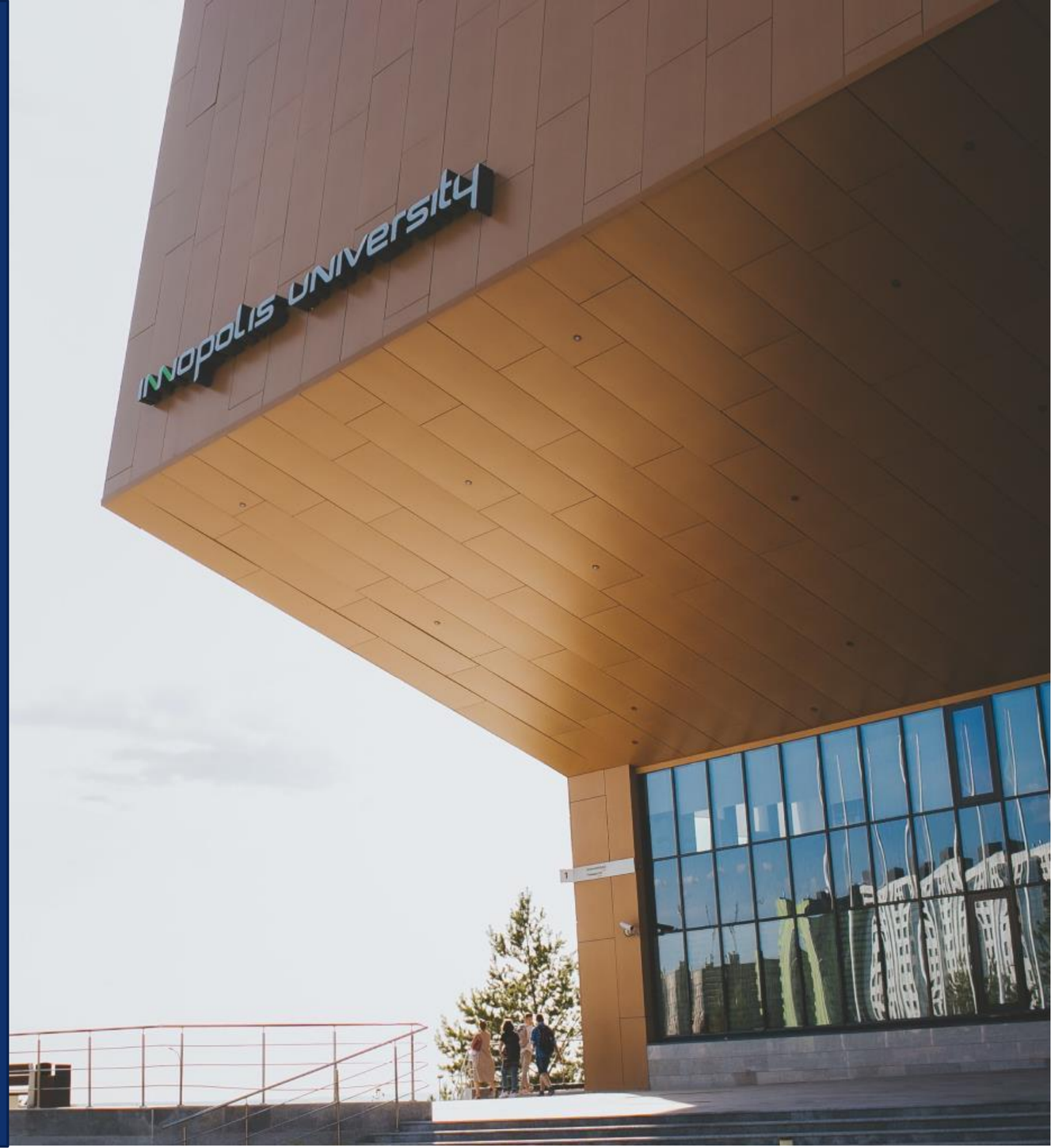
# Verilog HDL and open-source EDA Tools OpenLANE for ASIC Design

Andrey V. Zykov, PhD



# Microchip design

- Application specific integrated circuits (ASIC)
- Filed-programmable gate array (FPGA)

# FPGA

- Lookup tables (LUT)
- Programmable switches
- Additional resources (DSP blocks, neural blocks, CPU cores)

# Electronic Design Automation (EDA) tools

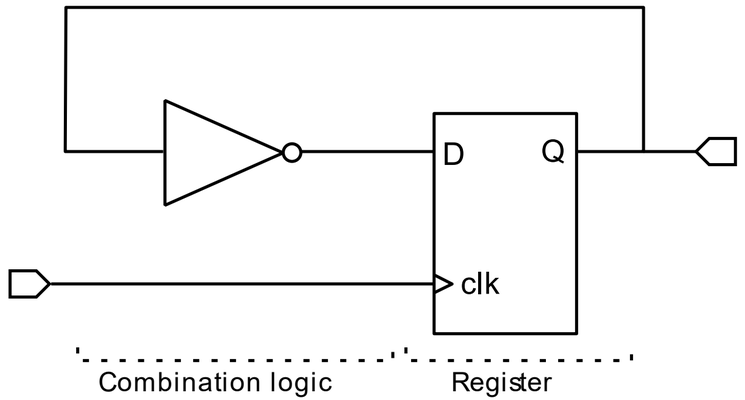
- Simulation
- Synthesis
  - High level synthesis (HLS)
  - Logic synthesis
  - Physical synthesis
- Analysis and Verification
- Optimization

# Verilog HDL

- Alternative language – VHDL
- Advanced version - SystemVerilog
- Synthesizable Verilog
- Behavioral Verilog

# Register-transfer level - RTL

- Design abstraction which models a synchronous digital circuit in terms of the flow of digital signals (data) between hardware registers
- Related to state machines



# Verilog: wires and registers

```
wire s_axi_bvalid;

reg clk;

reg  s_axi_bready = 0;  // initialized


wire [1:0]s_axi_bresp;

reg  [31:0]s_axi_wdata = 0;
```

# Verilog: modules

```
module sum(
    input  [31:0] a,
    input  [31:0] b,
    output [31:0] s
);
    assign s = a + b;
endmodule

module test();
    wire [31:0] x1, x2, res;
    sum sum_instance(.a(x1), .b(x2), .s(res));
endmodule
```

# Verilog: ALWAYS block

```
module counter(
    input  clk,
    input  reset,
    output reg [15:0] s
);
    always @(posedge clk) begin
        if(reset) begin
            s <= 16'h0000;
        end else begin
            s <= s + 1;
        end
    end
endmodule
```

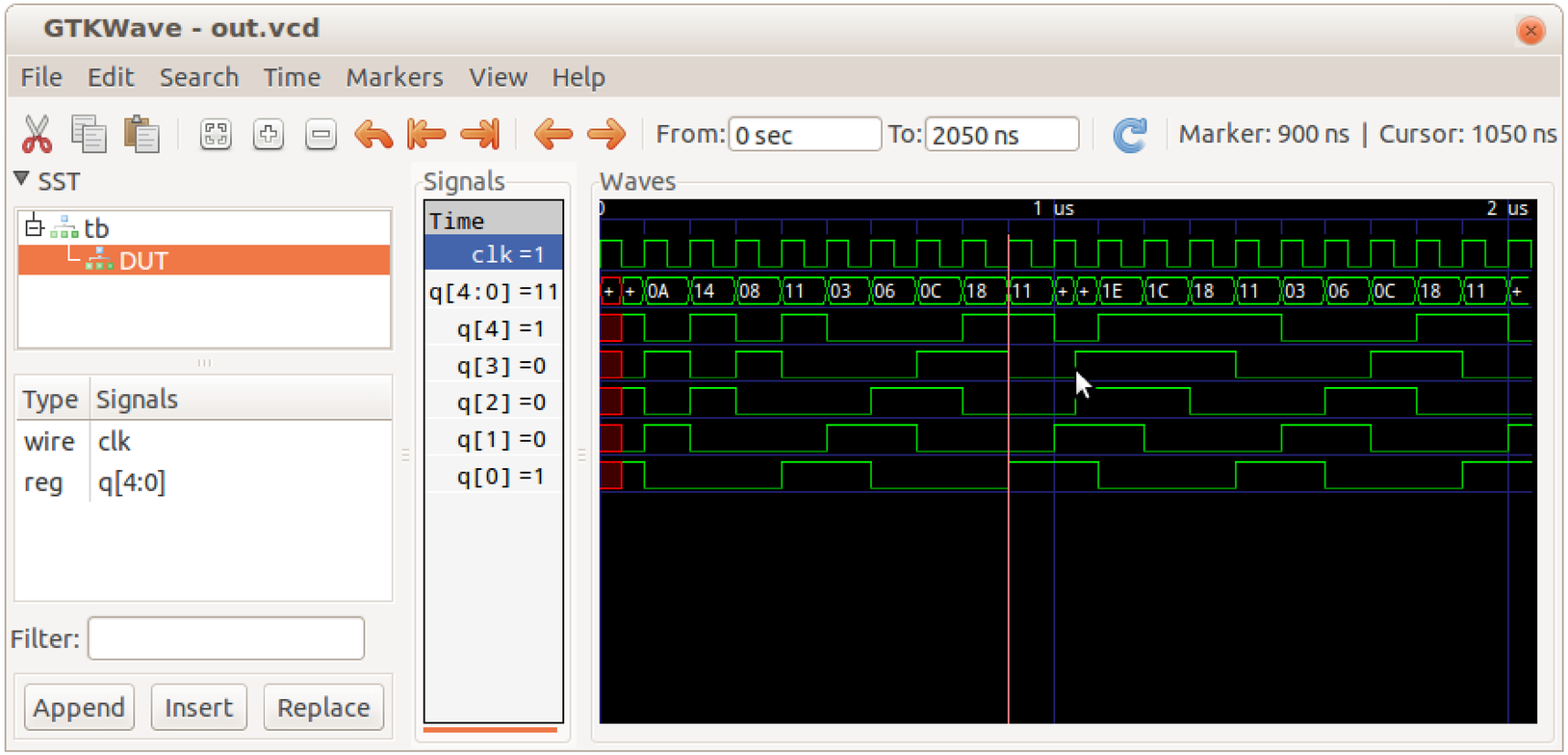
# Verilog: test bench

```
`timescale 1ns / 1ps
module top1( );
    reg clk, rst;
    wire [15:0] res
    counter i1(.clk(clk), .reset(rst), .s(res));

    initial begin
        clk  = 1'b1;
        rstn = 1'b1;
        #95         rst = 1'b0;
        #50000      $finish;
    end
    always #10 clk <= ~clk;
endmodule
```

# Simulator “Icarus Verilog”

- Open-source Verilog simulator “Icarus Verilog”
- Waveform visualization “GTKWave”



# Open-source motivation

- High cost of commercial EDA tools
- Not easy to extend functionality using you own algorithms
- Independence of foreign software vendors

# Open-source design flows

1. **ALLIANCE/CORIOLIS**
   LIP6, France
   1990-2000 years (currently in support state)

2. **Qflow**
   R. Timothy Edwards, Open Circuit Design
   since 2013 year

3. **OpenROAD**
   financed by DARPA
   started on Summer 2018

4. **OpenLANE**
   Efabless company
   since Fall 2019 year

# Most relevant open-source tools

1. **Yosys**
   synthesis and optimization of Verilog HDL

2. **ABC**
   library for logical circuit optimiztion

3. **Magic и KLayout**
   tools for geometry synthesis (GDSII)

4. **Netgen**
   layout versus schematic (LVS)

5. **CVC**
   circuit validity check

# OpenLANE: open-source design flow

1. **RTL-synthesis**
   Yosys, ABC

2. **Physical synthesis**
   OpenROAD

3. **Geometry sythesis**
   Magic and KLayout

4. **Physiscal verification**
   Netgen, CVC, Magic (for DRC)

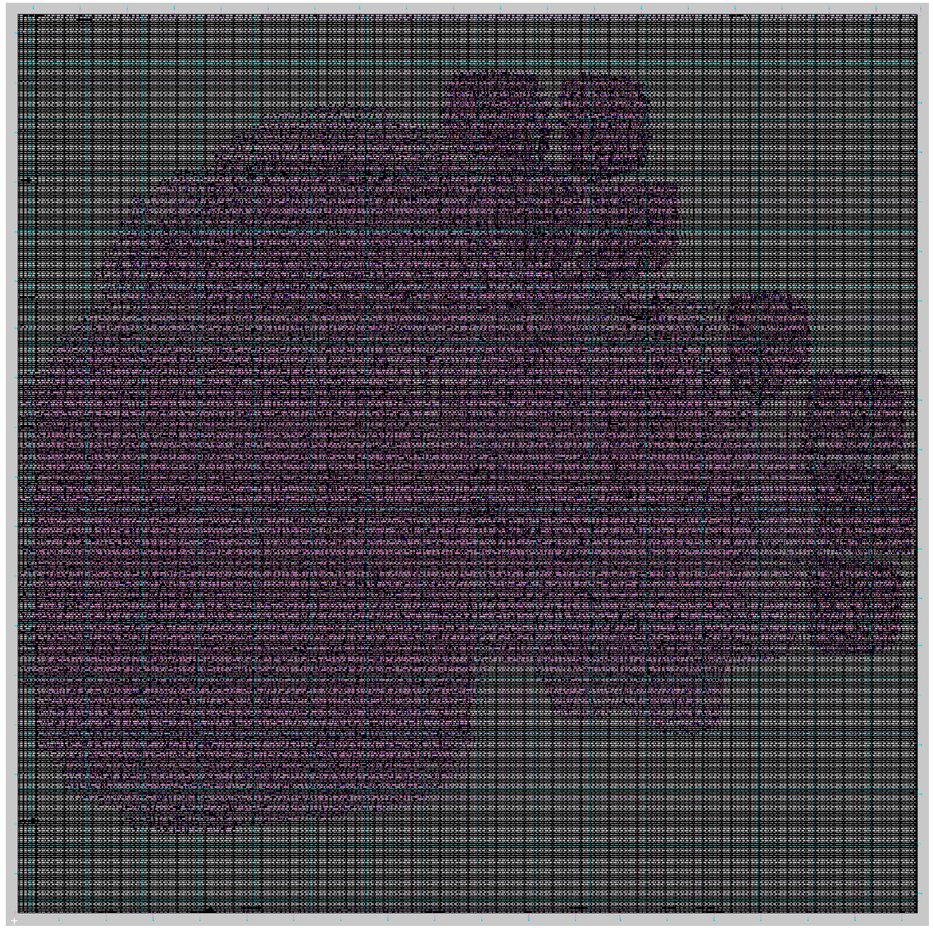
# Experiment

1. Priority arbiter modules (from 4 clients up to 96 clients)
2. Modules AES and AES-core OpenLANE test suit
3. Full synthesis in OpenLANE up to geometry in GDSII
4. Comparison of RTL-synthesis in OpenLANE (only Yosys part) and Cadence GENUS

# Example of geometry in Magic tool

Geometry GDSII for AES module

(square block 1.2x1.2 mm)



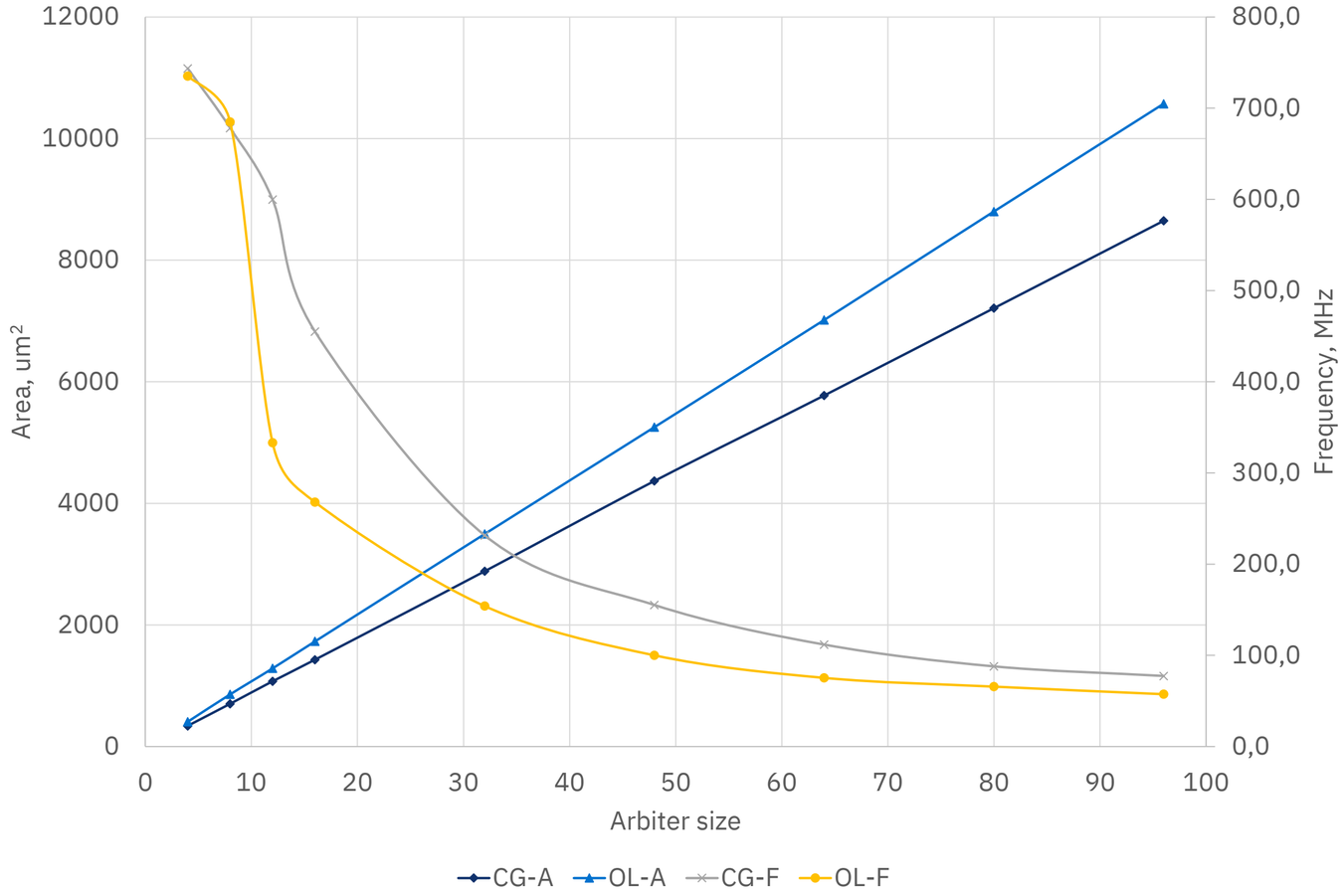
# Results of full synthesis in OpenLANE up to GDSII

| Modules | Area (um$^2$) | Cells count | Flip-flops count | Maximum frequency (MHz) |
|---|---|---|---|---|
| **arbiter 4** | 10138 | 36 | 12 | 515,5 |
| **arbiter 8** | 8551 | 74 | 24 | 436,7 |
| **arbiter 12** | 8551 | 115 | 36 | 401,6 |
| **arbiter 16** | 8134 | 153 | 48 | 337,8 |
| **arbiter 32** | 7647 | 308 | 96 | 185,2 |
| **arbiter 48** | 11418 | 469 | 144 | 119,5 |
| **arbiter 64** | 15258 | 619 | 192 | 88,9 |
| **arbiter 80** | 19335 | 782 | 240 | 73,7 |
| **arbiter 96** | 23332 | 943 | 288 | 63,4 |
| **AES** | 1519574 | 18753 | 2994 | 76,0 |
| **AES-core** | 837443 | 18720 | 2476 | 91,7 |

# RTL-synthesis in OpenLANE and in Cadence GENUS

| Modules | Area (um²) | Cells count | Flip-flops count | Maximum frequency (MHz) |
|---|---|---|---|---|
| arbiter 4 | 409 / 339 | 21 / 13 | 12 / 12 | 735,3 / 743,5 |
| arbiter 8 | 860 / 702 | 50 / 29 | 24 / 24 | 684,9 / 678,2 |
| arbiter 12 | 1286 / 1072 | 79 / 47 | 36 / 36 | 333,3 / 599,7 |
| arbiter 16 | 1730 / 1428 | 107 / 61 | 48 / 48 | 268,1 / 455,0 |
| arbiter 32 | 3497 / 2883 | 222 / 126 | 96 / 96 | 154,1 / 231,5 |
| arbiter 48 | 5254 / 4368 | 337 / 199 | 144 / 144 | 100,1 / 155,2 |
| arbiter 64 | 7015 / 5774 | 451 / 275 | 192 / 192 | 75,4 / 111,8 |
| arbiter 80 | 8800 / 7211 | 566 / 355 | 240 / 240 | 65,8 / 87,9 |
| arbiter 96 | 10571 / 8648 | 681 / 434 | 288 / 288 | 57,5 / 77,4 |
| AES | 228303 / 162288 | 17161 / 11993 | 2994 / 2987 | * / 144,5 |
| AES-core | 209968 / 169987 | 16881 / 14192 | 2476 / 2987 | * / 157,7 |

# RTL-synthesis of arbiter modules in OpenLANE and Cadence GENUS



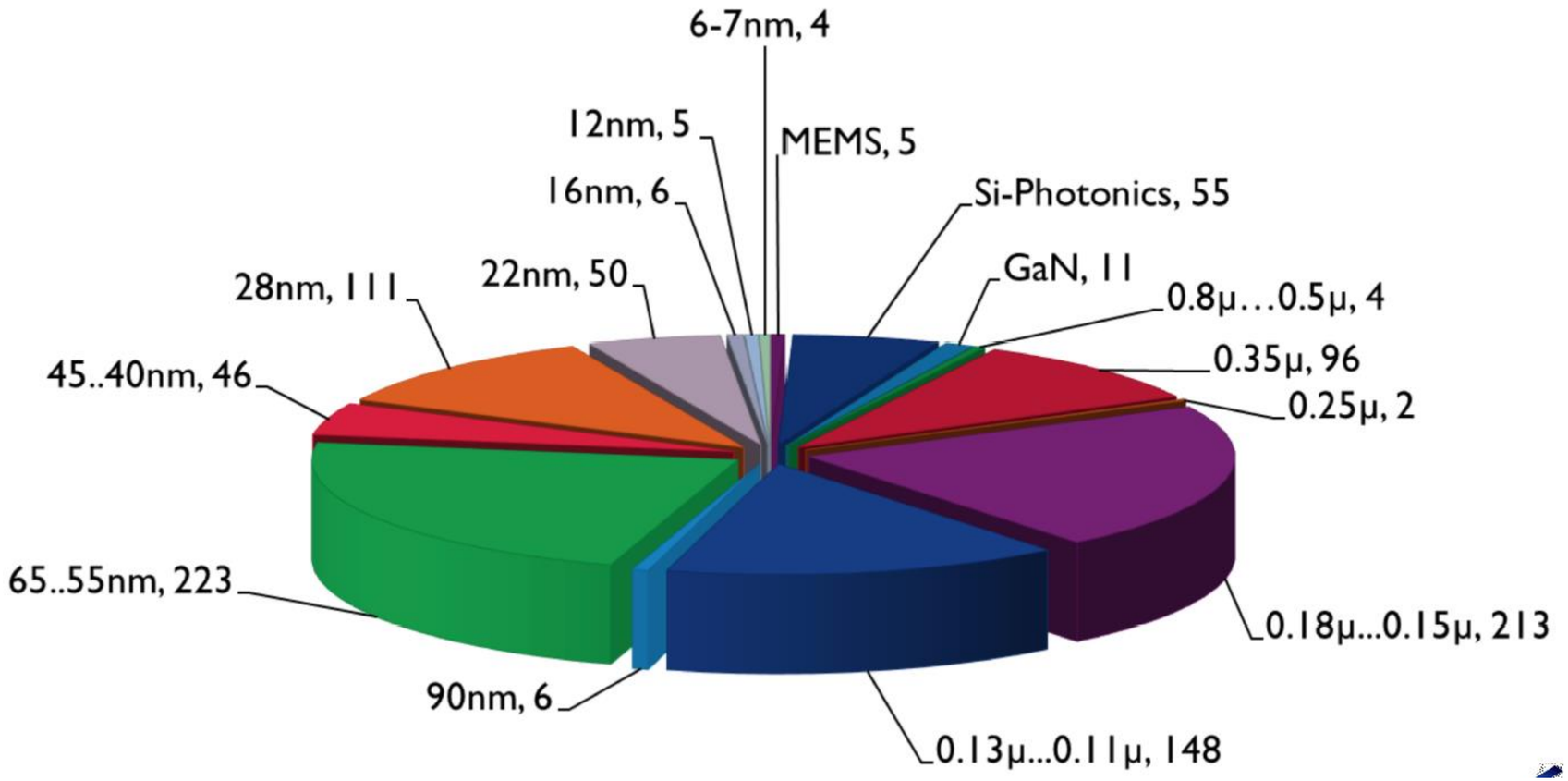
# Conclusion

1. OpenLANE – viable chose to design real microchip using open-source tools (demonstrated in chipIgnite program)

2. Though commercial tools demonstrate higher optimization level, the results of open-source tools are comparable

3. It is expected that OpenROAD will reach full operational state in the future and will replace OpenLANE

# Microchip technology



# Thank you!
# Questions?