Digital Design Labs and Project

Dr. Alexandre Levisse

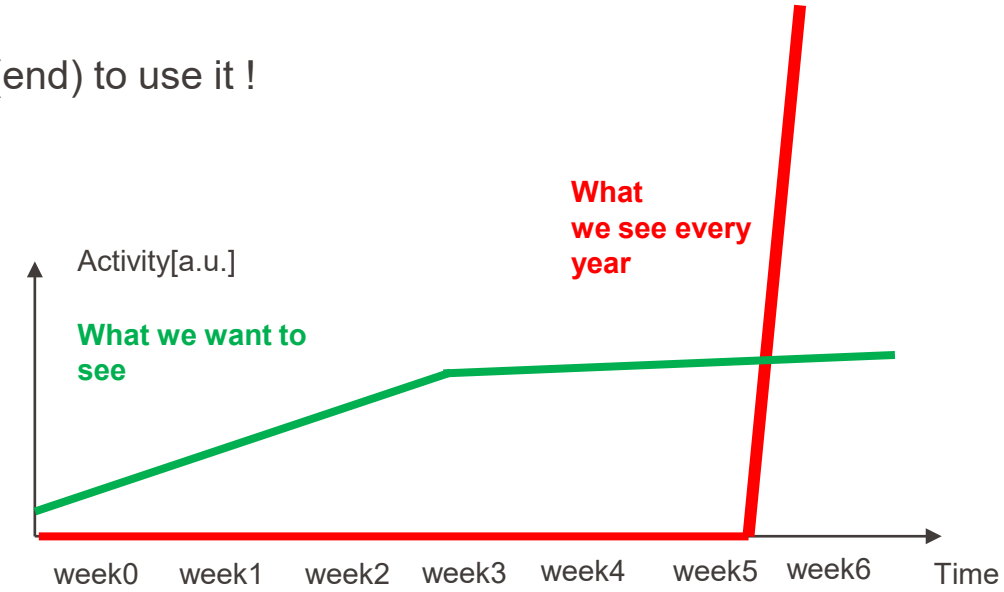
École polytechnique fédérale de Lausanne

EDAlabs 2024-2025

General Information

Alexandre Levisse

- Teaching assistants will be available for help and support.
  - Ask for help when you need some, but be mindful as you are a lot.
  - If the session is intense (i.e., TAs being extremely busy), and your question is not urgent, prioritize the forum.
- Use the forum !
  - **USE IT**
  - Don’t wait for the last week(end) to use it !



Deadlines management

Alexandre Levisse

- You will always have to deal with deadlines in your life
- Deadline management is not innate

8 tips to meet your deadlines

1. Communicate a clear deadline
2. Break down the project
3. Have a start and completion date for each step
4. Block off time on your calendar
5. Focus on action (vs. motion)
6. Communicate progress with your team
7. Add a buffer time
8. Don’t overcommit

8 Tips to Meet Deadlines Without Over-Stressing Yourself

Here are 8 best practices to set realistic deadlines and meet them... without feeling stressed or overwhelmed.

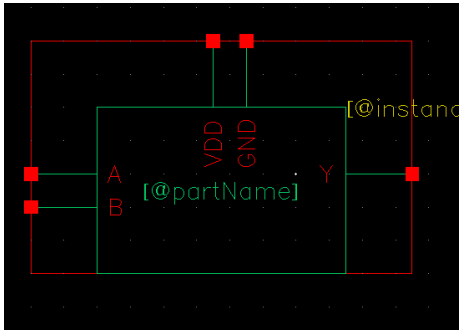
Outline

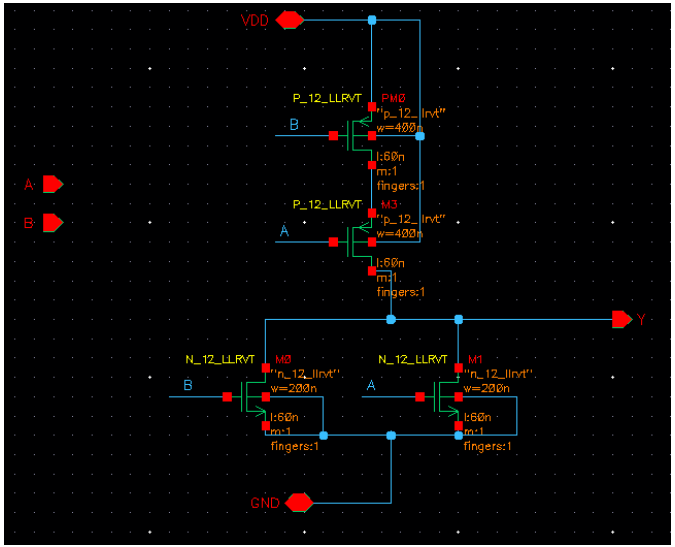
- What did you do so far?
- Why digital design ?
- How ?
- What about the lab ?

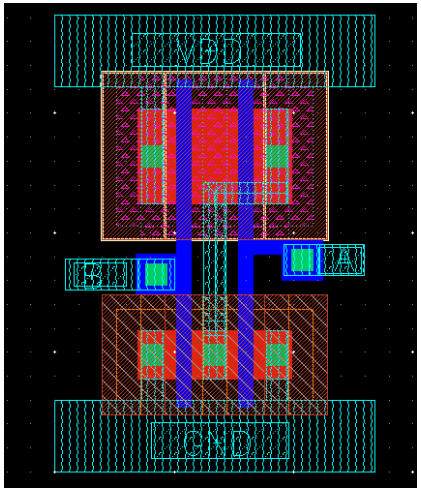
Alexandre Levisse

What did you do so far ?

- Design a small logic block
- E.g. of a NOR2 Gate





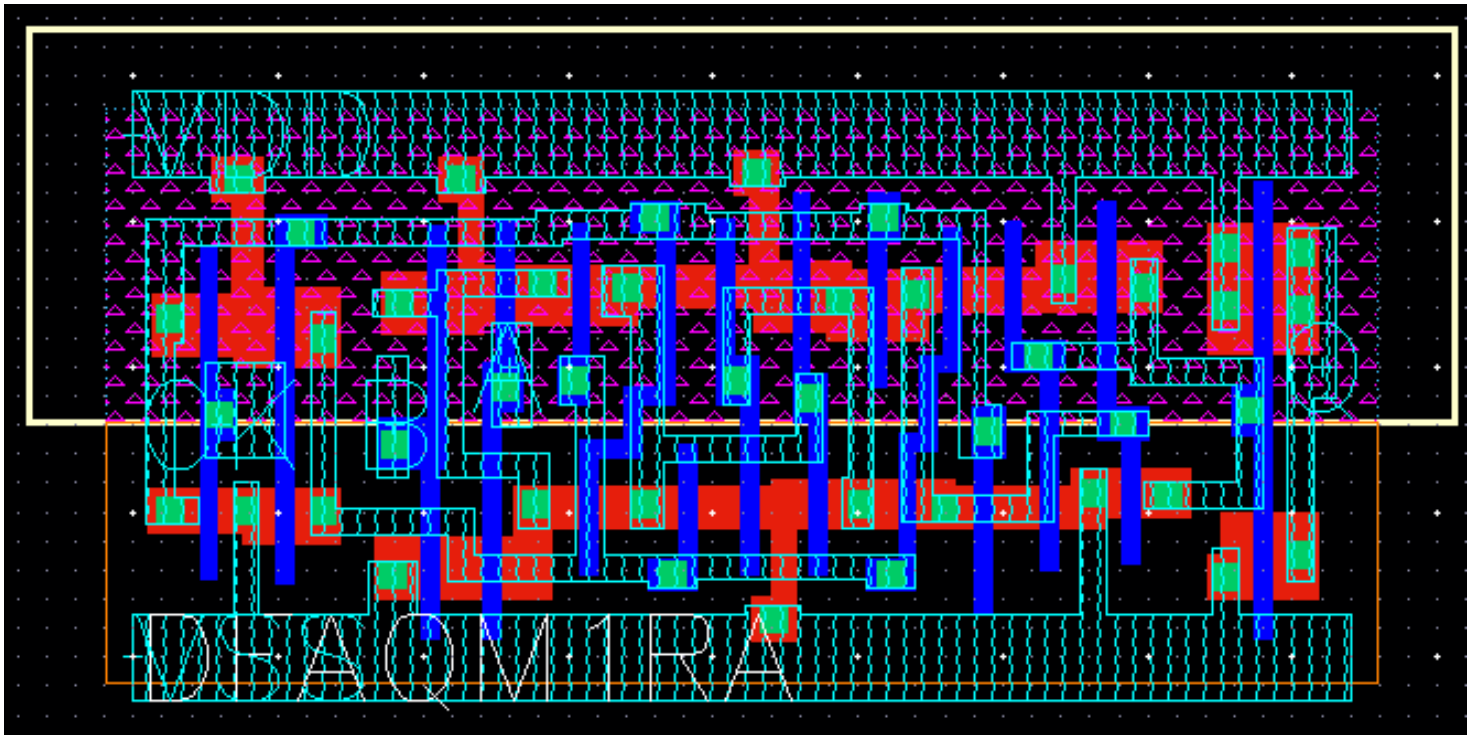


What did you do so far ?

- Size transistors
- Assemble logic gates together
- Make a structured and hierarchical design
- Make a floorplan
- Route signals
- Manage power lines
- Manage well biasing

- Your design :
  - Around 700 MOS.
  - Standard cell-based design
  - No variability analysis
  - No temperature analysis
  - No power optimization

E.g. of an optimized layout



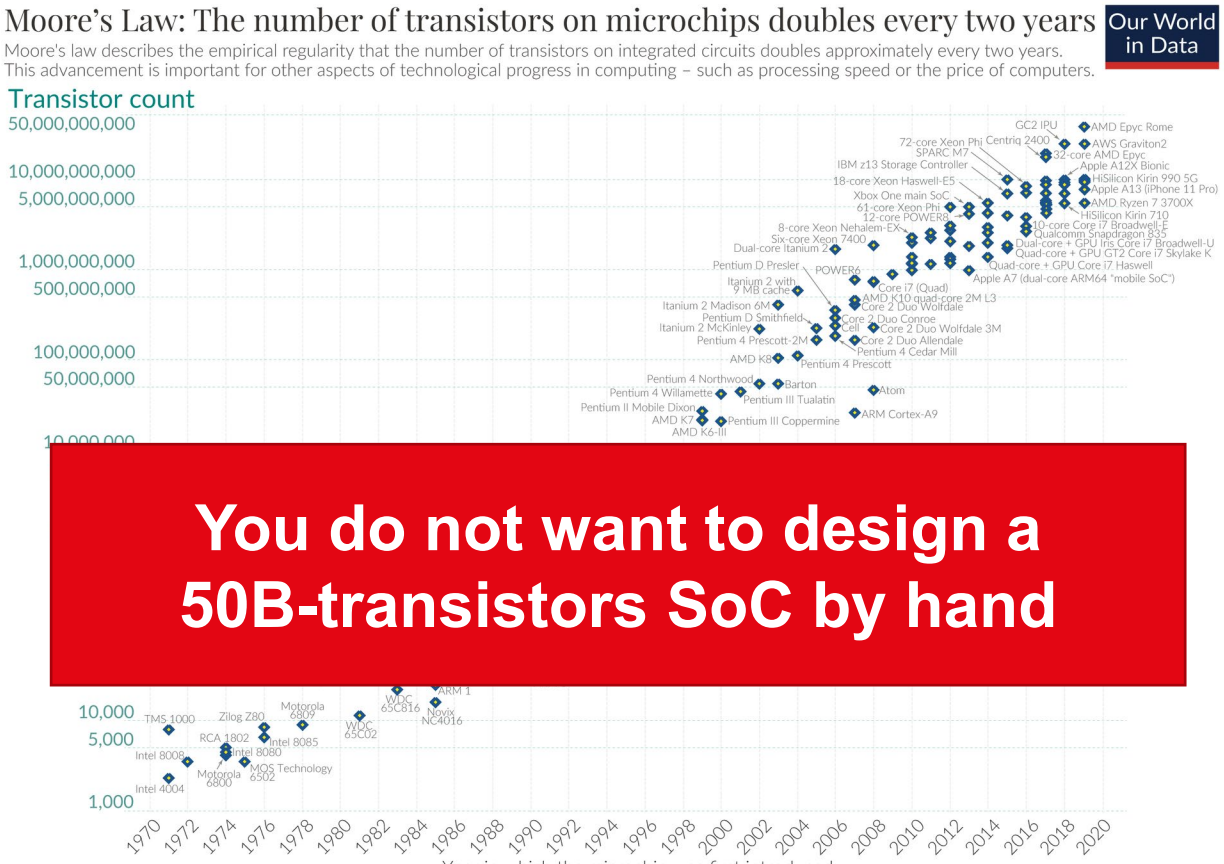
What did you do so far ?

- It took you a lot of efforts
- Reason 1 : This was most likely your first design
- Reason 2 : Full-Custom design takes time and effort

- You need a good reason to do a Full-Custom design
- One reason being : “I have no other way to design these”
  - Voltage management
  - Converters (ADCs/DACs)
  - Frequency generators
  - Memories
  - Radio Frequency
  - Specific digital IPs

You want to automate everything when you can do so.
How to automate standard cell based design?

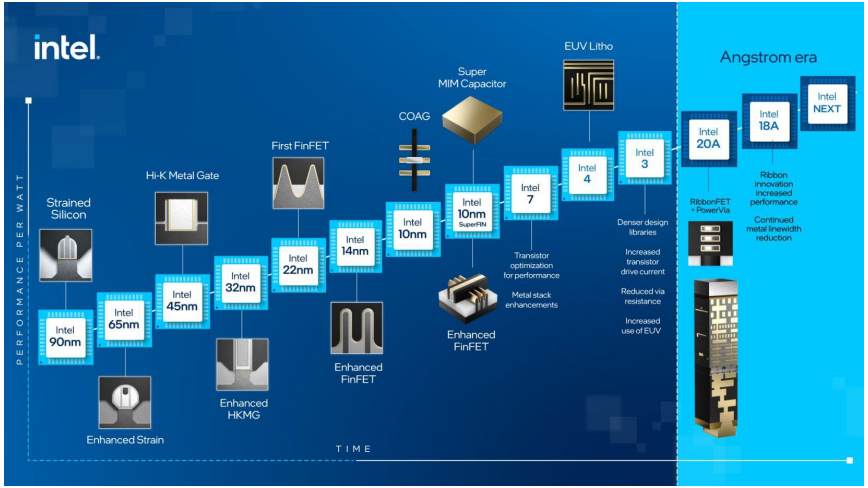
Why Digital Design ?

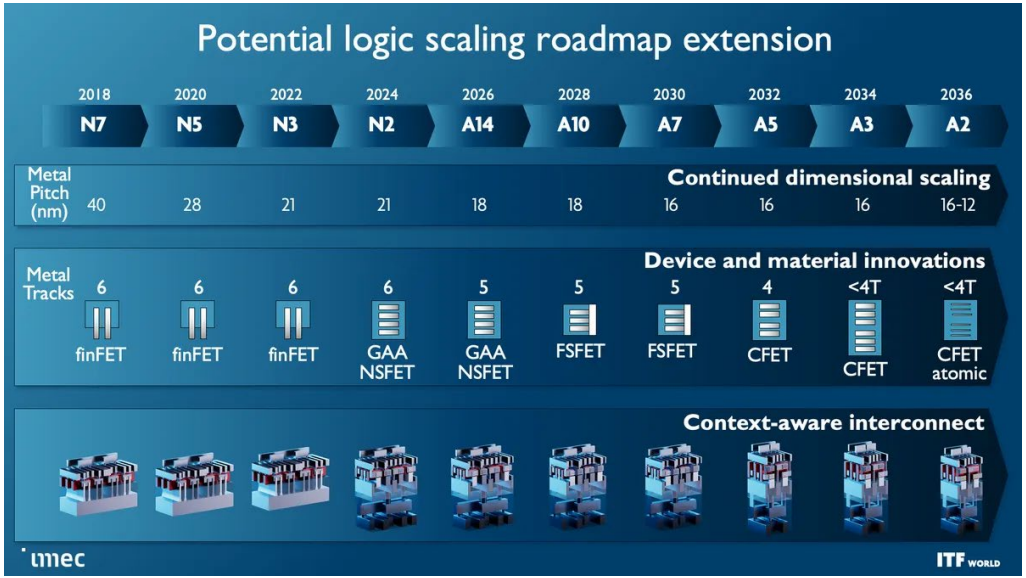


You do not want to design a 50B-transistors SoC by hand

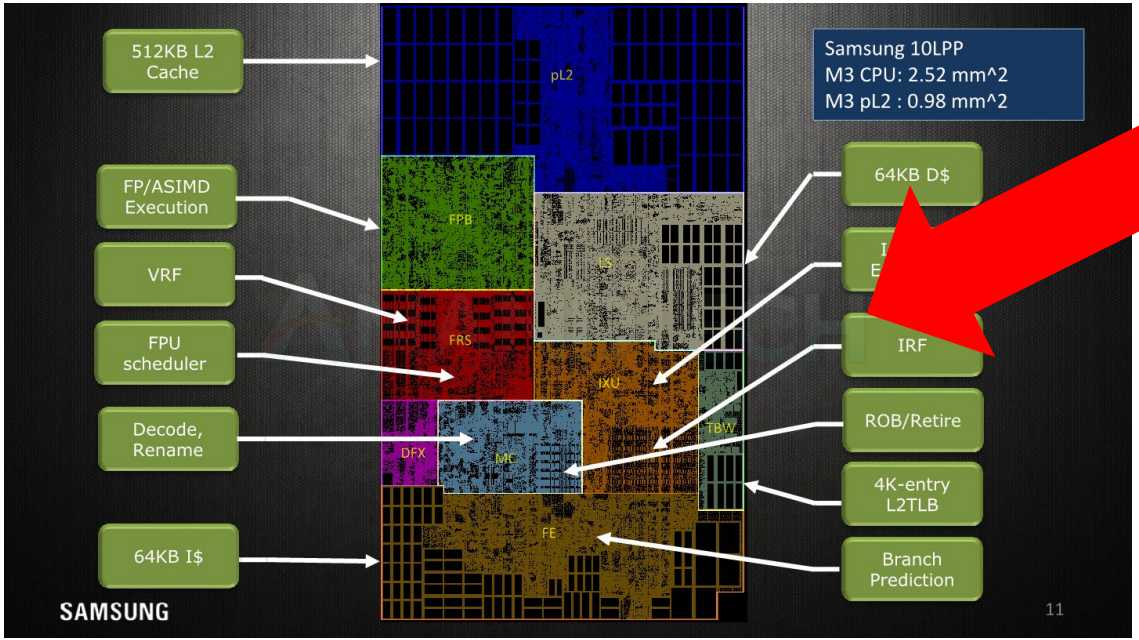
Why Digital Design ?

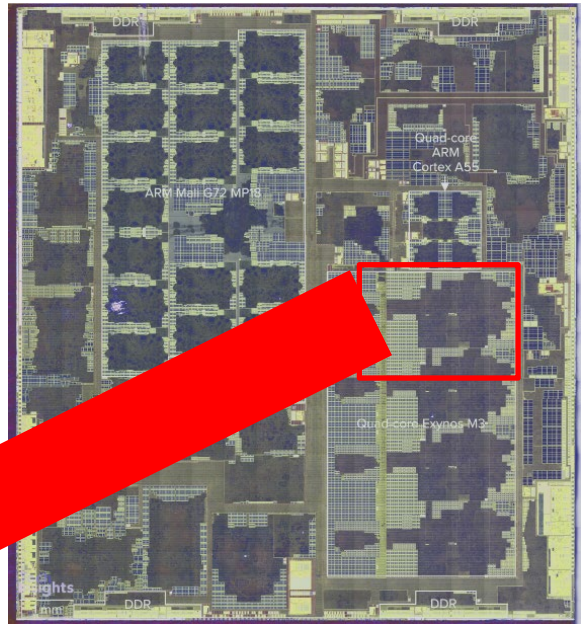
- Scaling is not going to stop



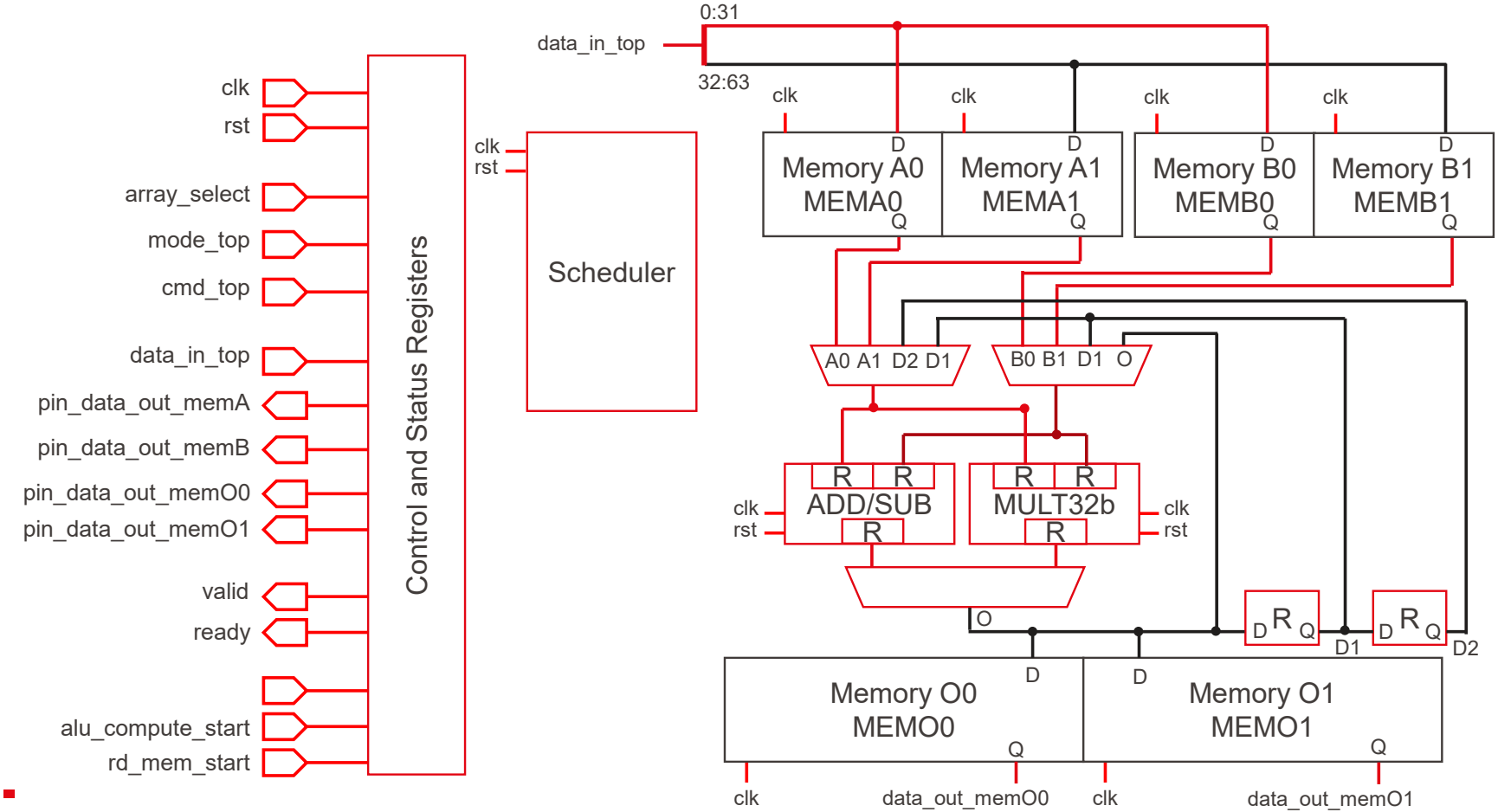


Hierarchical designs





Design Example for this lab



How do you do that ?

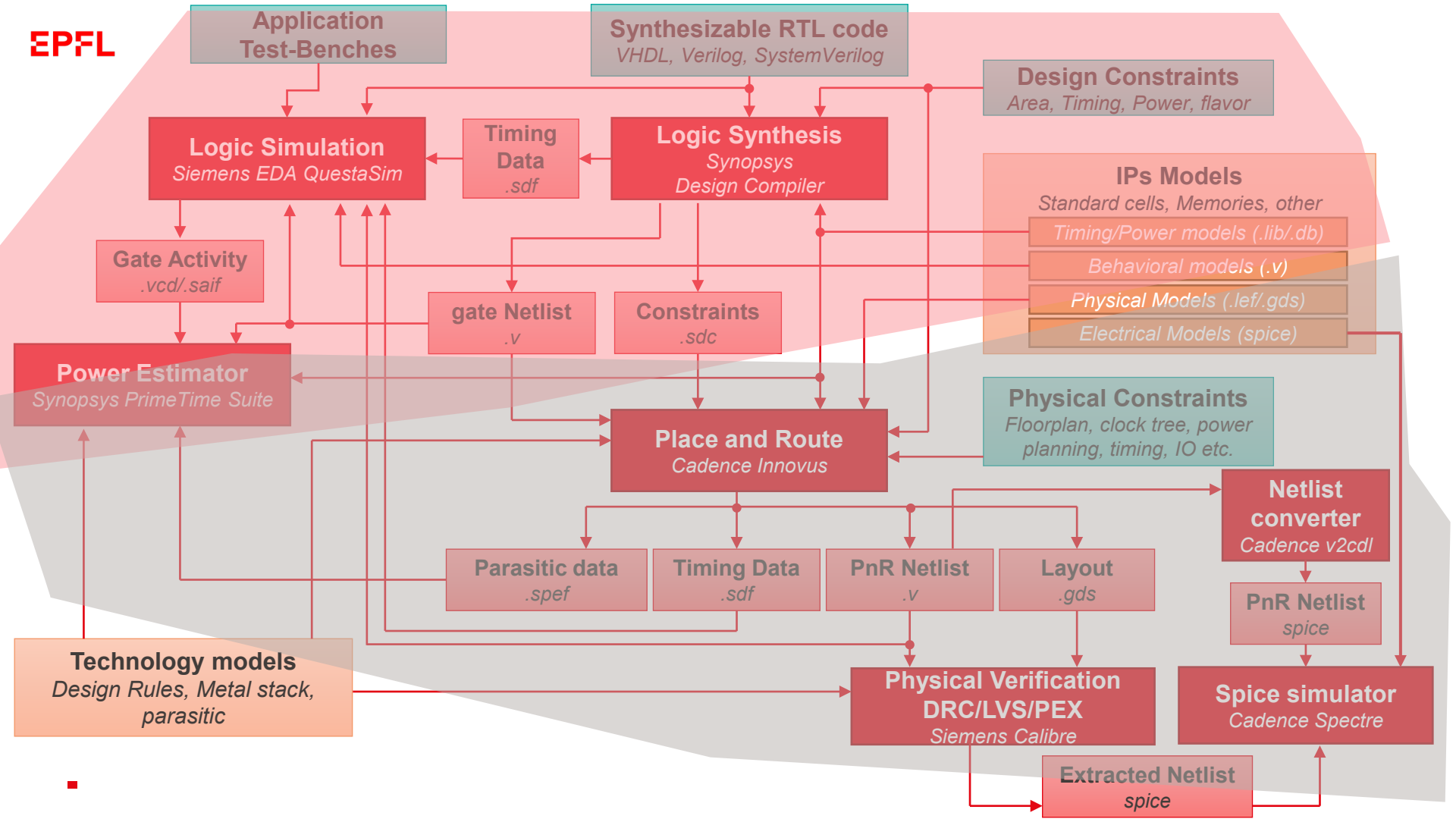
- Start by writing some code describing the circuit you are trying to design

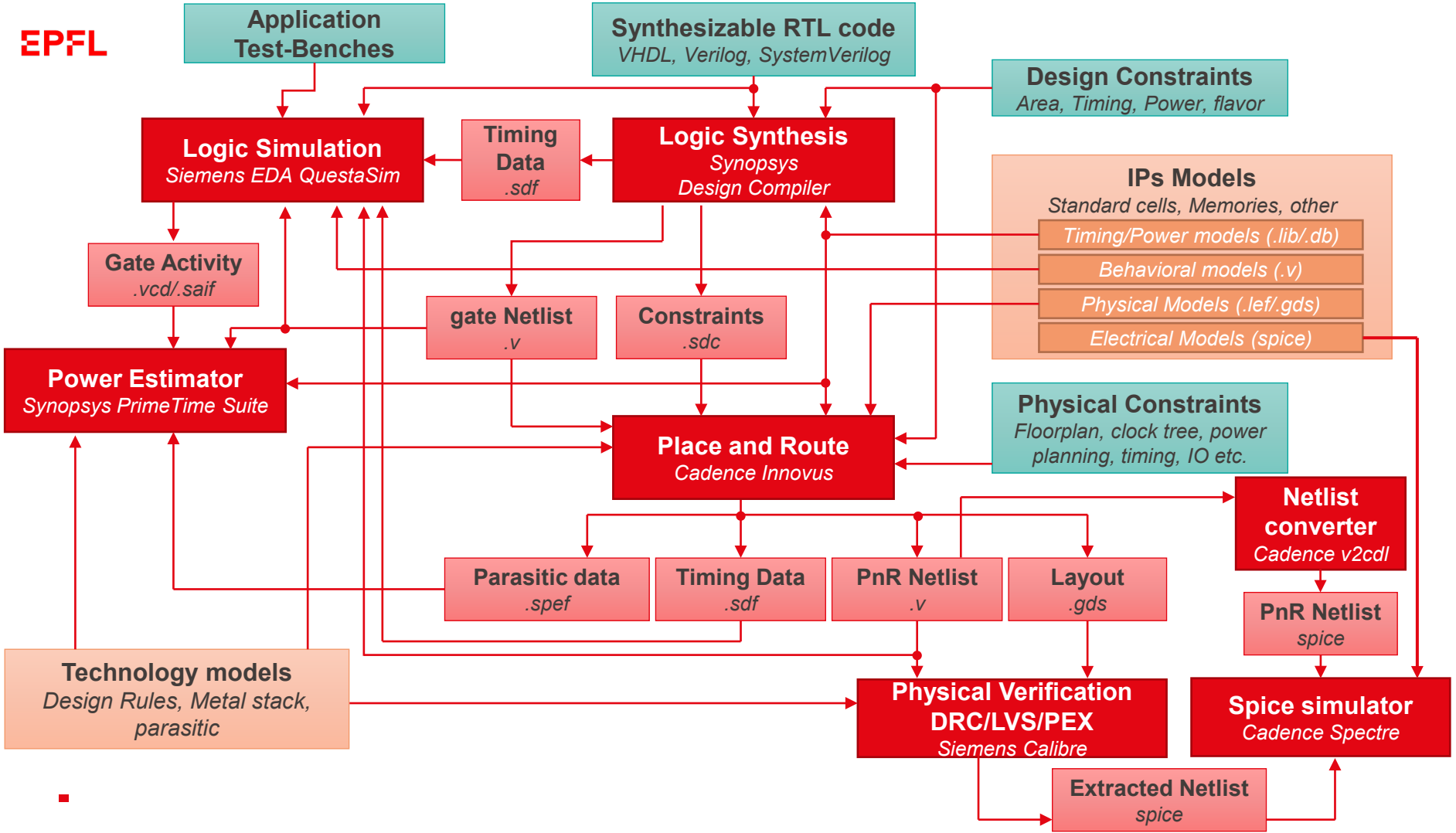
```
21
22  library IEEE;
23  use IEEE.STD_LOGIC_1164.ALL;
24  use ieee.numeric_std.all;
25  use work.alu32_pkg.all;
26
27  entity mult_32b is
28      Port ( op1 : in STD_LOGIC_VECTOR (31 downto 0);
29             op2 : in STD_LOGIC_VECTOR (31 downto 0);
30             res : out STD_LOGIC_VECTOR (63 downto 0);
31             done : out STD_LOGIC;
32             cmd : in STD_LOGIC_VECTOR (1 downto 0);
33             clk : in STD_LOGIC;
34             rst : in STD_LOGIC;
35             start : in STD_LOGIC);
36  end mult_32b;
37
38  architecture rtl of mult_32b is
39
40  signal reg_op1 : std_logic_vector (31 downto 0);
41  signal reg_op2 : std_logic_vector (31 downto 0);
42
43  begin
44
45  INPUT_REGS : process (rst, clk)
46    begin
47        if rst = '0' then
48            if rising_edge(clk) then
49                -- do stuff
50                reg_op1 <= op1;
51                reg_op2 <= op2;
52
53            end if;
54        else
55            -- reset signals
56            reg_op1 <= x"00000000";
57            reg_op2 <= x"00000000";
58        end if;
59    end process;
60
61      CLOCKED_MULTIPLIER : process (rst, clk)
62      begin
63          if rst = '0' then
64              if rising_edge(clk) then
65                  if start = '1' then
66                      case cmd is
67                          when ALU_MULT =>
68                              res <= std_logic_vector(unsigned(reg_op1) * unsigned(reg_op2));
69                              done <= '1';
70                          when others =>
71                              res <= (others => '0');
72                              done <= '0';
73                      end case;
74                  else
```

How do you do that ?

- Then write some stimulus to test that your circuit behaves the way it is supposed to behave

```
101
102     wait for 2.5*T;
103     rst          <= '0';
104     wait for 20.5 * T;
105     global_en <= '1';
106
107  -- writing 16 words to array 0
108     wait for 5*T;
109     array_select  <= ARRAY_SEL0;
110     data_in_top   <= X"0000000000000000";
111     wait for T;
112     wr_mem_start <= '1';
113     wait for 2*T;
114     wr_mem_start <= '0';
115     data_in_top   <= X"000000000000000F";  -- add
116     wait for 2*T;
117     data_in_top   <= X"0000000000000000";  -- mult
118     wait for 2*T;
119     data_in_top   <= X"0000FFFF00000000";  -- add
120     wait for 2*T;
121     data_in_top   <= X"00000000FFFFFFFF";  -- sub
122     wait for 2*T;
123     data_in_top   <= X"00000000FFFFFFFF"; --noop
124     wait for 2*T;
125     data_in_top   <= X"00000000FFFFFFFF"; --mult
126     wait for 2*T;
127     data_in_top   <= X"000000000000000F";
128     wait for 2*T;
129     data_in_top   <= X"00000000000000FF";
130     wait for 2*T;
131     data_in_top   <= X"0000000000000FFF";
132     wait for 2*T;
133     data_in_top   <= X"0000000000FFFFFF";
134     wait for 2*T;
135     data_in_top   <= X"00000000FFFFFFF";
136     wait for 2*T;
137     data_in_top   <= X"00000000FFFFFFFF";
138     wait for 2*T;
139     data_in_top   <= X"00000000FFFFFFFF";
140     wait for 2*T;
141     data_in_top   <= X"00000000FFFFFFFF";
142       wait for 2*T;
143     data_in_top   <= X"00000000FFFFFFFF";
144       wait for 2*T;
145     data_in_top   <= X"0000000000000000";
146     wait for 30*T;
```


EPFL
Application Test-Benches
Synthesizable RTL code
VHDL, Verilog, SystemVerilog
Design Constraints
Area, Timing, Power, flavor
Logic Simulation
Siemens EDA QuestaSim
Timing Data
.sdf
Logic Synthesis
Synopsys Design Compiler
IPs Models
Standard cells, Memories, other
Timing/Power models (.lib/.db)
Behavioral models (.v)
Physical Models (.lef/.gds)
Electrical Models (spice)
Gate Activity
.vcd/.saif
gate Netlist
.v
Constraints
.sdc
Power Estimator
Synopsys PrimeTime Suite
Physical Constraints
Floorplan, clock tree, power planning, timing, IO etc.
Place and Route
Cadence Innovus
Netlist converter
Cadence v2cdl
Parasitic data
.spef
Timing Data
.sdf
PnR Netlist
.v
Layout
.gds
PnR Netlist
spice
Technology models
Design Rules, Metal stack, parasitic
Physical Verification DRC/LVS/PEX
Siemens Calibre
Spice simulator
Cadence Spectre
Extracted Netlist
spice

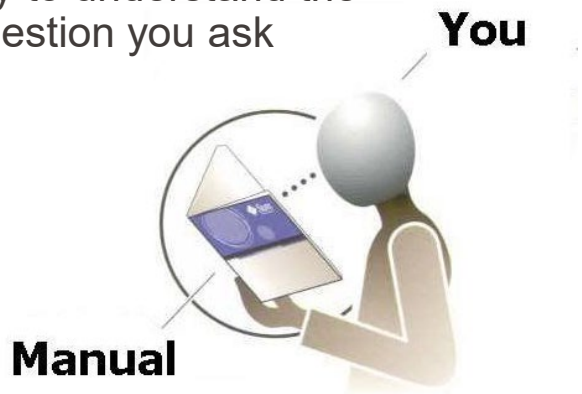

EPFL
Application Test-Benches
Synthesizable RTL code
VHDL, Verilog, SystemVerilog
Design Constraints
Area, Timing, Power, flavor
Logic Simulation
Siemens EDA QuestaSim
Timing Data
.sdf
Logic Synthesis
Synopsys Design Compiler
IPs Models
Standard cells, Memories, other
Timing/Power models (.lib/.db)
Behavioral models (.v)
Physical Models (.lef/.gds)
Electrical Models (spice)
Gate Activity
.vcd/.saif
gate Netlist
.v
Constraints
.sdc
Power Estimator
Synopsys PrimeTime Suite
Physical Constraints
Floorplan, clock tree, power planning, timing, IO etc.
Place and Route
Cadence Innovus
Netlist converter
Cadence v2cdl
Parasitic data
.spef
Timing Data
.sdf
PnR Netlist
.v
Layout
.gds
PnR Netlist
spice
Technology models
Design Rules, Metal stack, parasitic
Physical Verification DRC/LVS/PEX
Siemens Calibre
Spice simulator
Cadence Spectre
Extracted Netlist
spice

Schedule

Session No	Date	Content	Assignments
1	20 Nov	Introduction, Logic Simulation, Logic Synthesis, Power Analysis, Design Space Exploration	
2	27 Nov		
3	28 Nov		
4	5 Dec		Report 1 : Front End December 10th 8PM
5	11 Dec	Memory Compilers Place and Route Power Analysis Post PnR verification Design Space Exploration	
6	12 Dec		
7	18 Dec		
8	19 Dec		Report 2 : Back End December 30th 8PM

Problems ?

- Step 1 :
  - Make sure you did follow the tutorial correctly
  - Read the errors and warnings, and try to understand them
  - Try to understand the question you ask



- Step 2 :
  - Clearly identify the problem
  - Contextualize it
  - Call a TA

Things to be careful about

- **DO NOT GET LATE !**

- All you need is in the tutorial. Go through it once. Understand everything that is going on in there before jumping in the project.
- Some tasks are tedious. We do them once by hand, and THEN we automate

- Follow Carefully the tutorial and everything should go well
- There are tons of information in these documents. Consider these as baseline reference documents for when you need to design something in the future.
- This presentation covers most of the points from the preamble – *still read it anyway at home !* It gives a good overview of the labs and context.

GRADING !

- QUESTIONS along the documents are NOT GRADED
- The grading will be based on your projects report and the success you have in:
  - Defining a plan
  - Applying it
  - Performing the analysis
  - Results quality

GRADING !

- **All the Deadlines are HARD, there will be no additional delay**
  - **A late project = 0**

- All the report submission moodle pages will be opened from the beginning of the project

- Front end : December 10th 8pm
- Back end : December 30th 8pm

Merci

Dr. Alexandre Levisse

École polytechnique fédérale de Lausanne