EITF35: Introduction to Structured VLSI Design

Part 7.1.1: Wrap Up

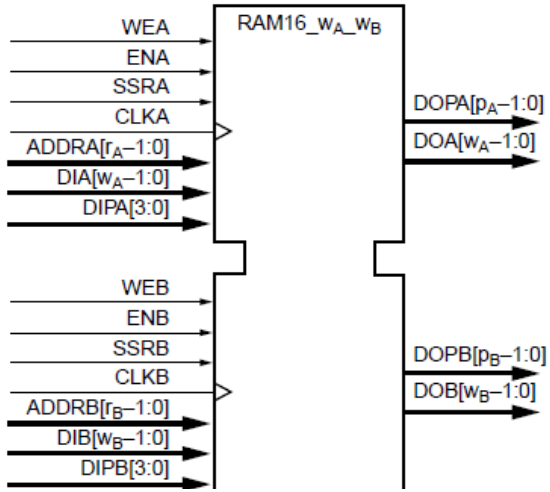
Liang Liu
liang.liu@eit.lth.se

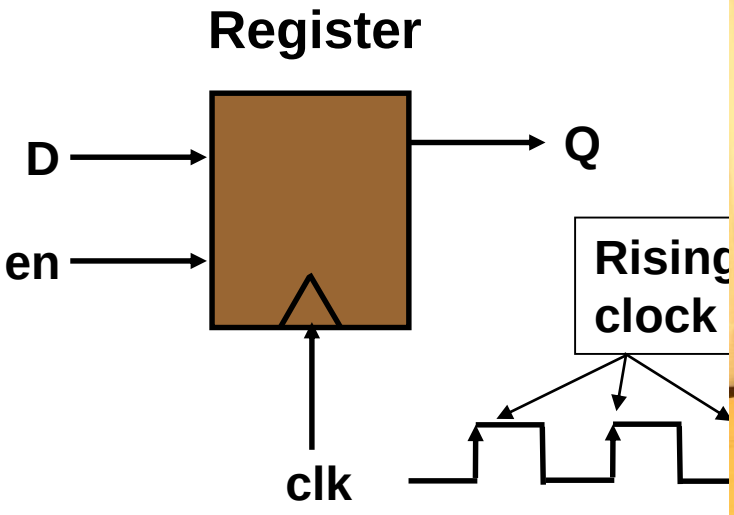
Outline

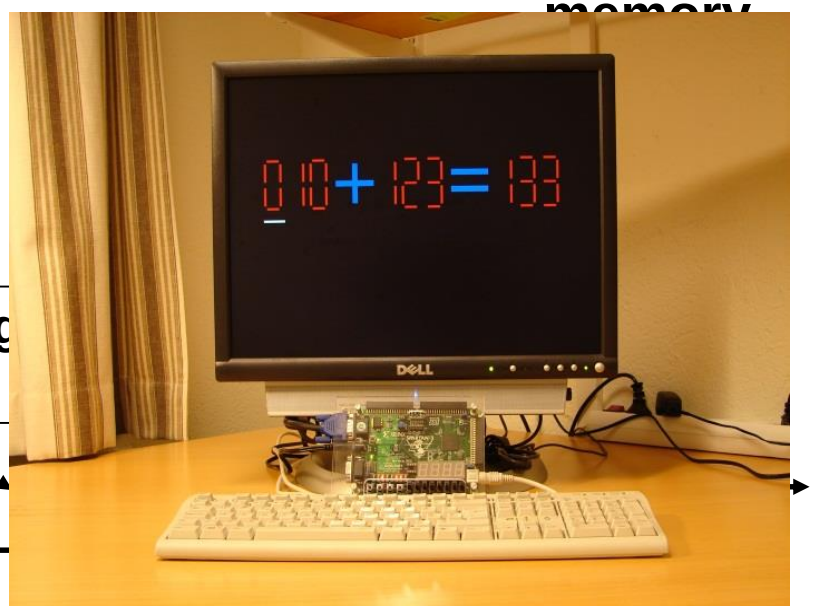
- Conclusion
- Next Step

What we have learned

- **Basic Building Blocks**
  - Combinational logic
  - Registers
  - Memory
- **Construct VLSI Systems**

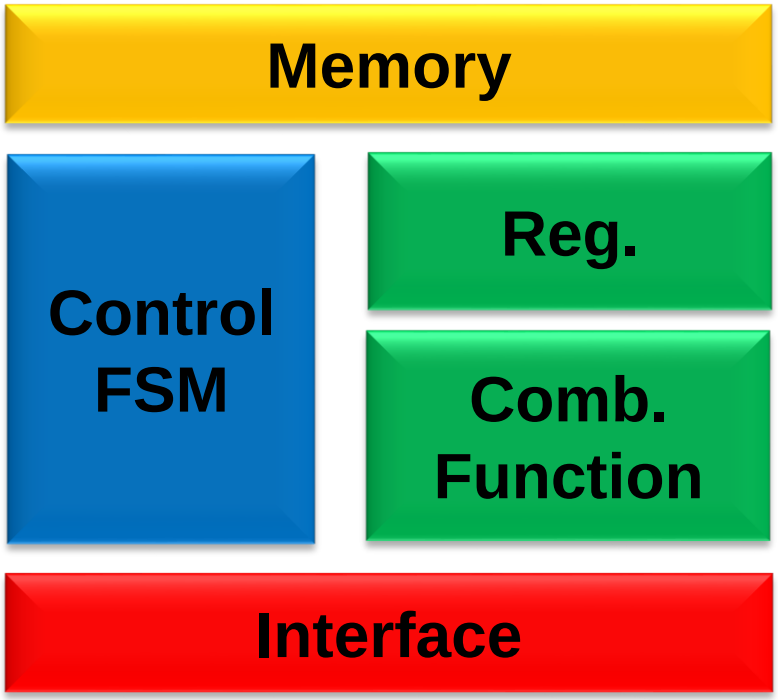






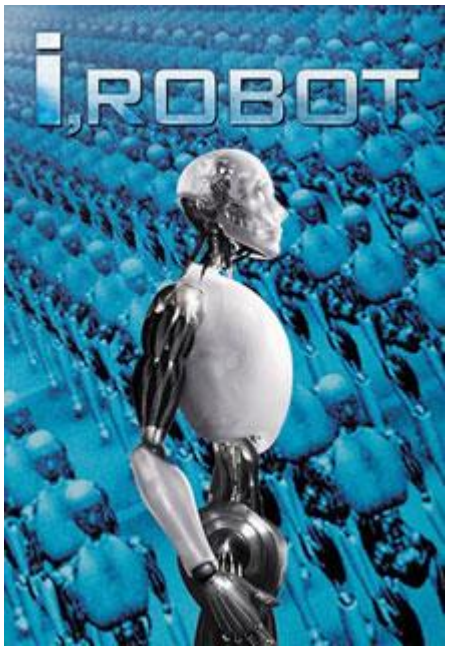
What, How, When

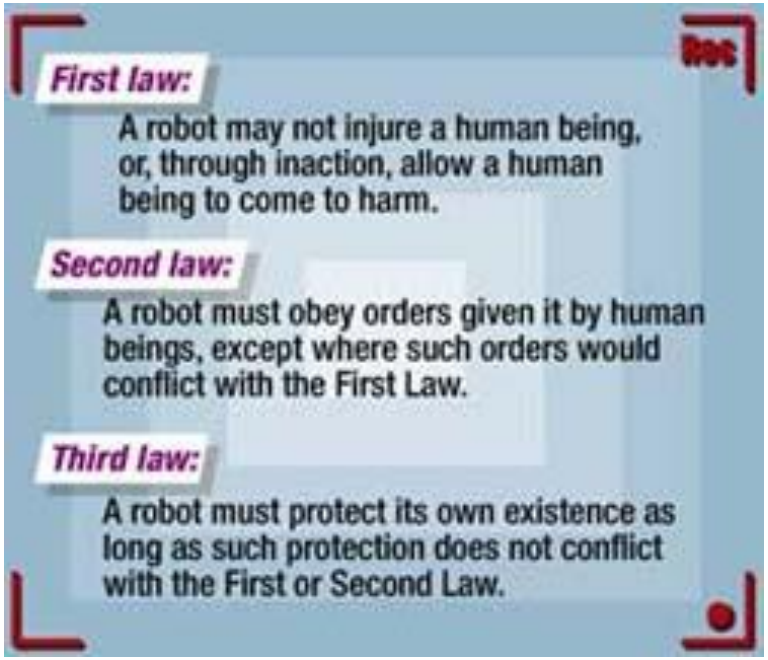
What we have learned



- **Assign.1**
  - Sequence Detector
- **Assign.2**
  - Keyboard controller
- **Assign.3**
  - ALU
- **Assign.4&5**
  - "Micro Processor"

VLSI ‘Laws’





Working hardware that meets the design requirement

Suggestion 1 – Design Flow

Specification

- **Understand the requirement**
  - Functionality & performance
  - Time to deadline

M = mod(X,Y) returns X - n.*Y, where n = floor(X./Y)

R = rem(X,Y) returns X - n.*Y, where n = fix(X./Y)

Rem or Mod?

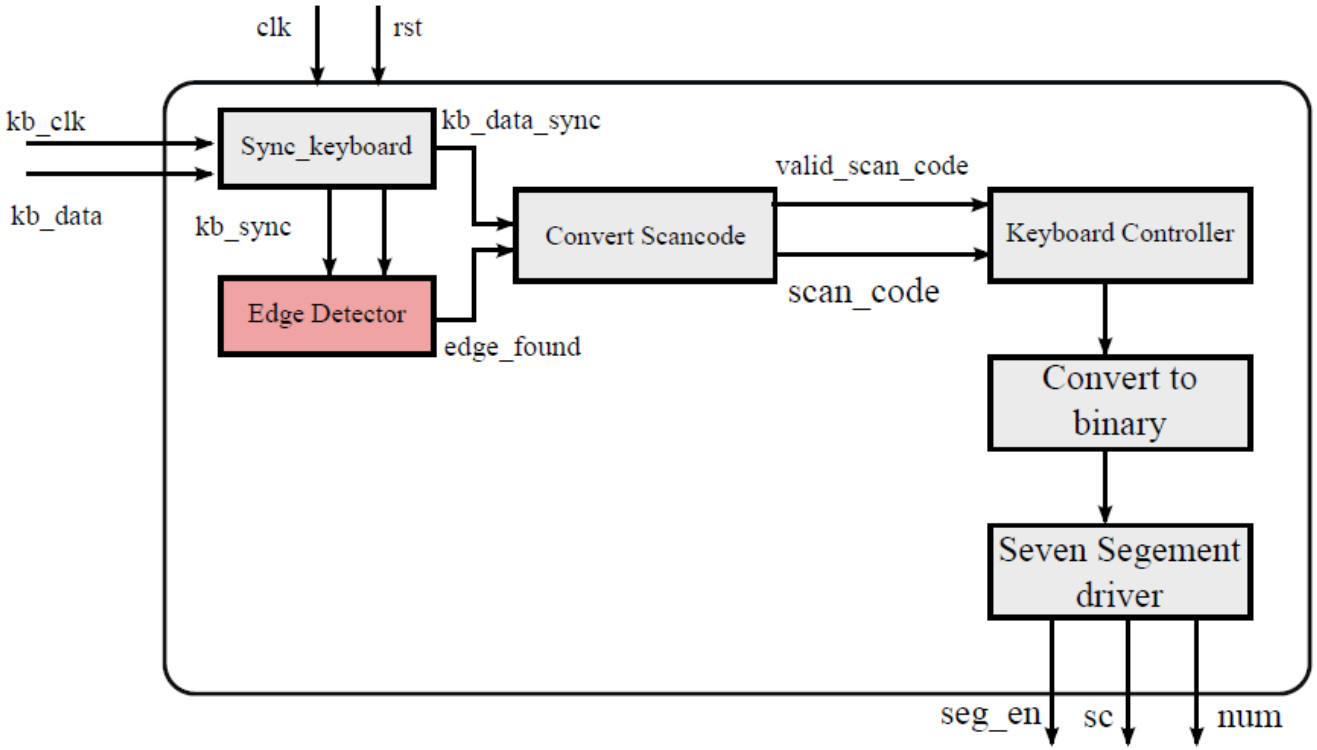
Mod(-100,3)=2
Rem(-100,3)=-1

Suggestion 1 – Design Flow

Specification

Block Diagram

- **Understand the requirement**
  - Functionality & performance
- **Draw a block diagram**



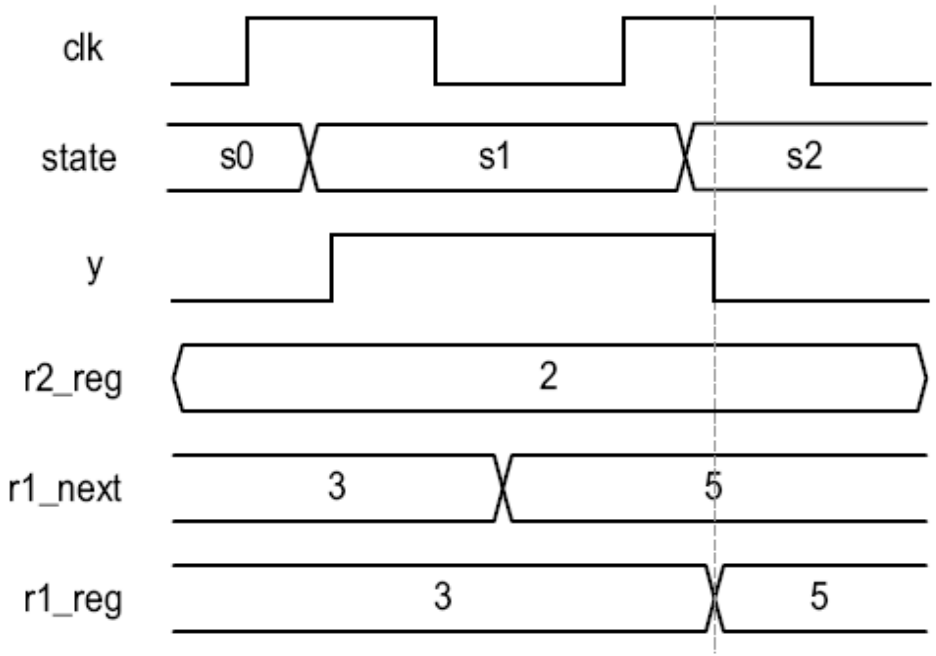
Suggestion 1 – Design Flow

Specification

Block Diagram

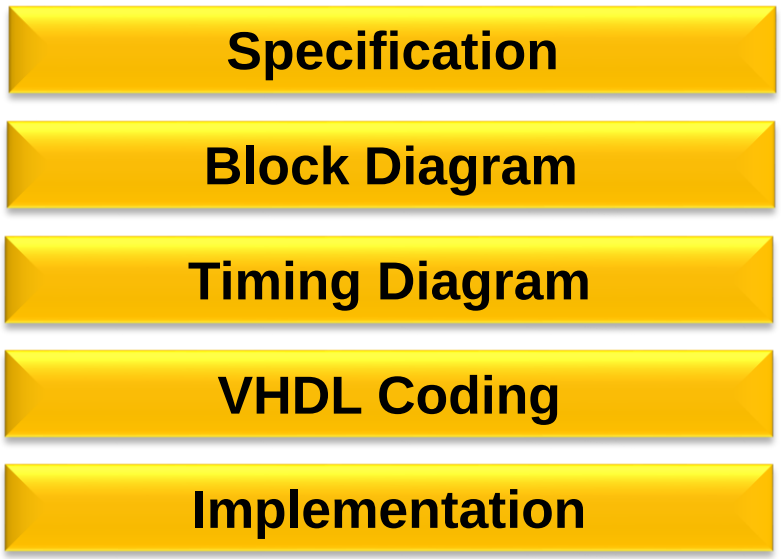
Timing Diagram

- **Understand the requirement**
  - Functionality & performance
- **Draw a block diagram**
- **Understand the timing**



(c) Timing diagram

Suggestion 1 – Design Flow



- **Understand the requirement**
  - Functionality & performance
- **Draw a block diagram**
- **Understand the timing**
- **VHDL is only a way to ask for component**
- **Implement your design**

Suggestion 2 – Verification

- **Verification is strictly required at each design stage**
- **Verify as much as possible, especially at early stage**
  - Simulate before implementation
  - Verify blocks before integration

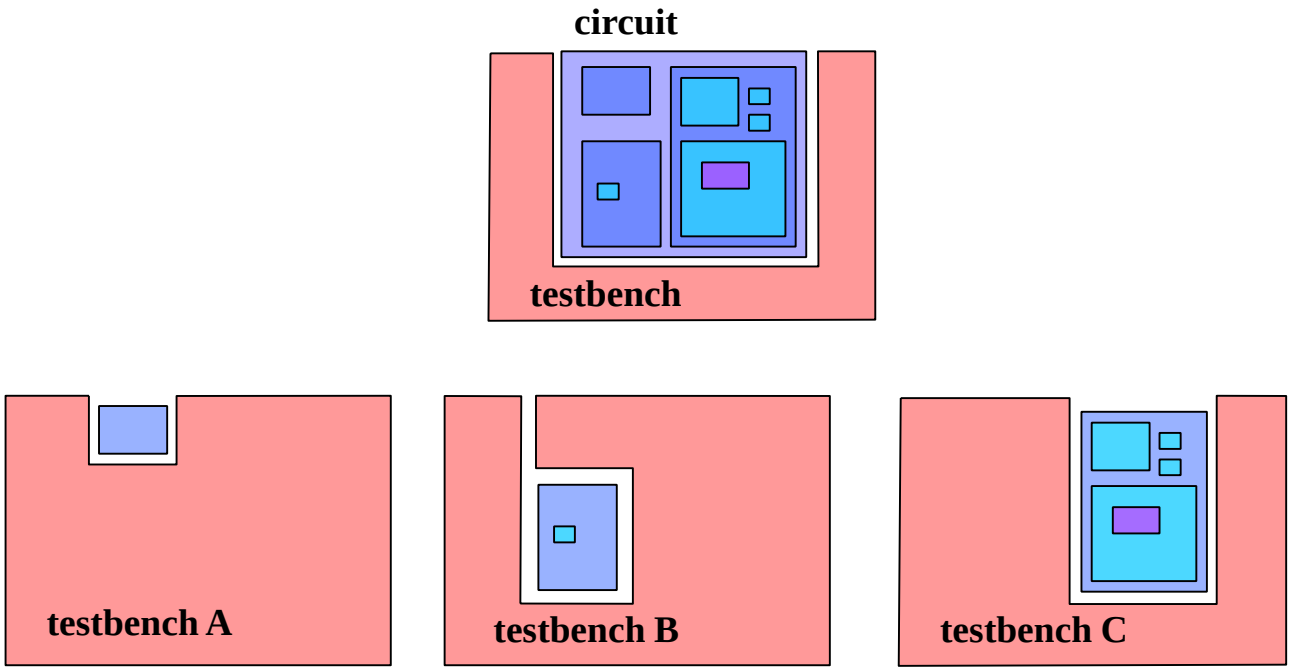
The cost of fixing a bug grows exponentially with the stage!!!

Suggestion 3 – Debug

- **There MUST be an error!**
  - Small & simple

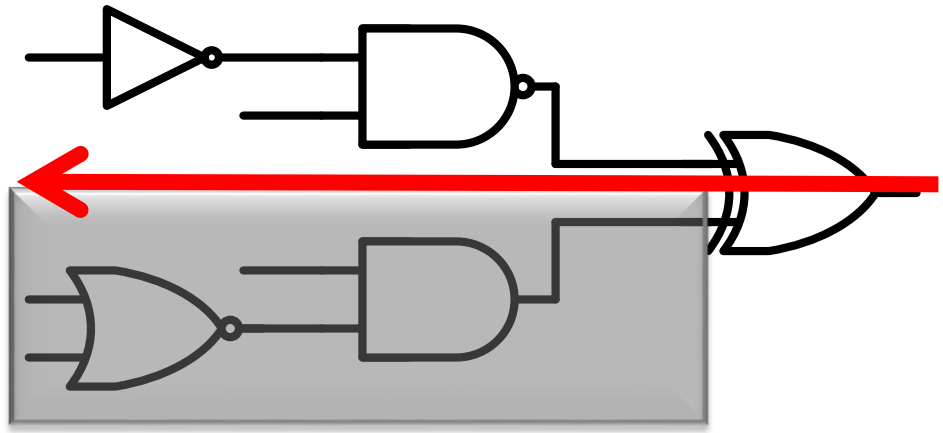
Suggestion 3 – Debug

- **There MUST be an error**
  - Small & simple
- **Locate the error**
  - Divide and conquer



Suggestion 3 – Debug

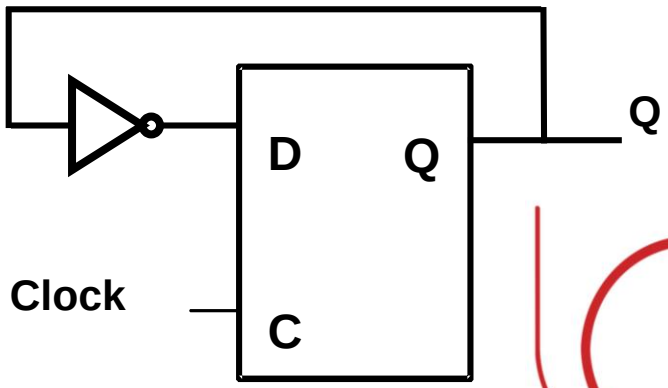
- **There MUST be an error**
  - Small & simple
- **Locate the error**
  - Divide and conquer
  - Error isolation
  - Trace back
  - Special stimuli

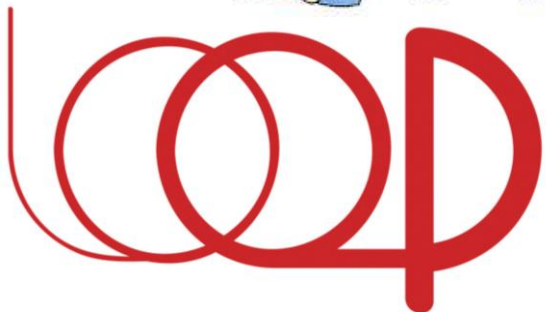


Suggestion 4 – Synthesizable VHDL

- **RTL simulation pass ≠ implementation success**
- **Typical un-controllable codes**
  - Latches
  - Mixed reg. with comb.
  - “wait” & “after”
  - Division…
  - Loop & variable

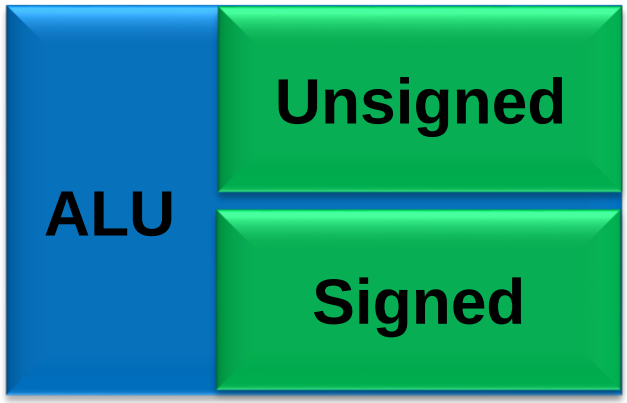
```
process(a,b)
  begin
    c<=c+a+b;
  end process;
```





Suggestion 5 – Hierarchical Design

- **Hierarchical design**
  - **Divided-and-conquer** strategy
  - Divide a system into ***smaller parts***
  - Constructs each module ***independently***



Suggestion 6 – Optimize at early stage

	Opt.	Ave.	Raw
Speed	200MHz	80MHz	20MHz
Area (Slice)	75	190	310

Suggestion 6 – Optimize at early stage

Algorithm

Architecture

Function Block

Implementation

```
x mod n = x - n·floor(x/n)
```

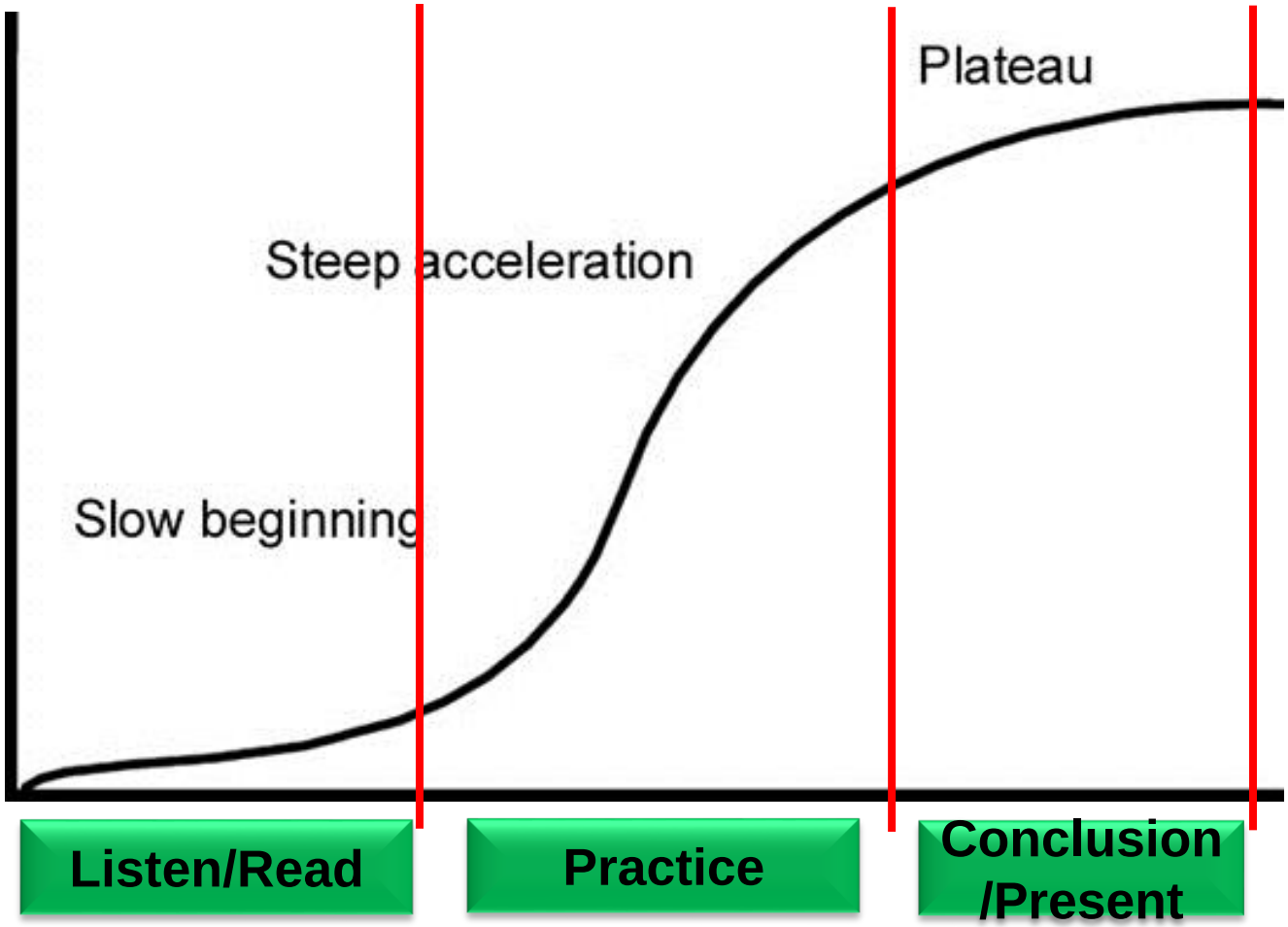
```
x = (x >> 6) + (x & 0x3f);
x = (x >> 4) + (x & 0xf);
x = (x >> 2) + (x & 0x3);
x = (x >> 2) + (x & 0x3);
if (x == 3) x = 0;
```

	Slices	4-input LUT
8-bit Multiplier	37	70
8-bit Adder	4	8

Suggestion 7 – Design Management

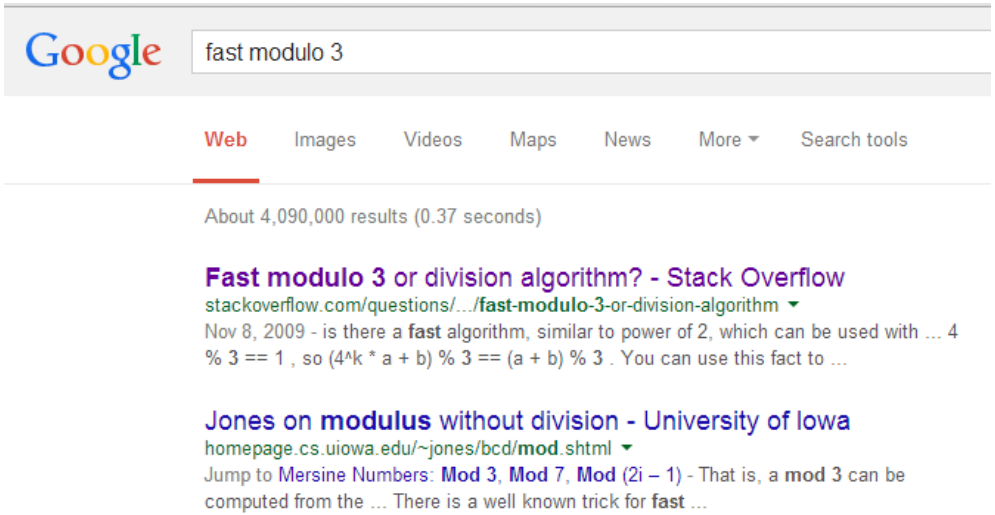
- **Folder**
  - Project_name (A3_ALU)
  - Algorithm, RTL, Sim, Synthesis, P&R…
  - Synthesis: Netlist, Script, Report, Readme.txt…
- **File Name**
  - "stage_design"
  - Design module or entity ("m_alu" or "ent_alu")
  - Test bench ("tb_alu")
- **Signal Name**
  - "direction_function_feature"
  - Pos-edge clock (i_clk_p)
  - Active low reset (i_rst_n)
- **Comments**
  - At least 30% of the codes

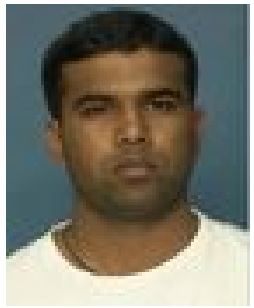
Suggestion 8 – Practice makes perfect



Suggestion 9 – Resource finding

http://www.eit.lth.se/course/eitf35





Rakesh Gangarajaiah

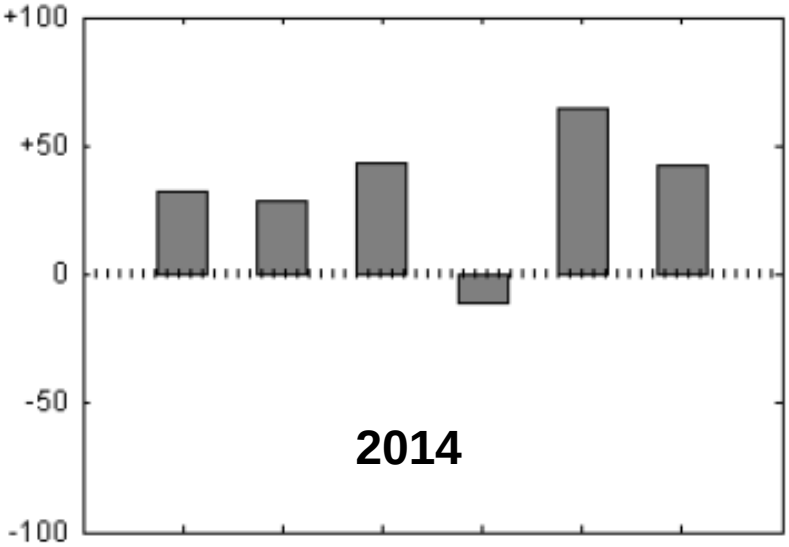
Mojtaba Mahdavi

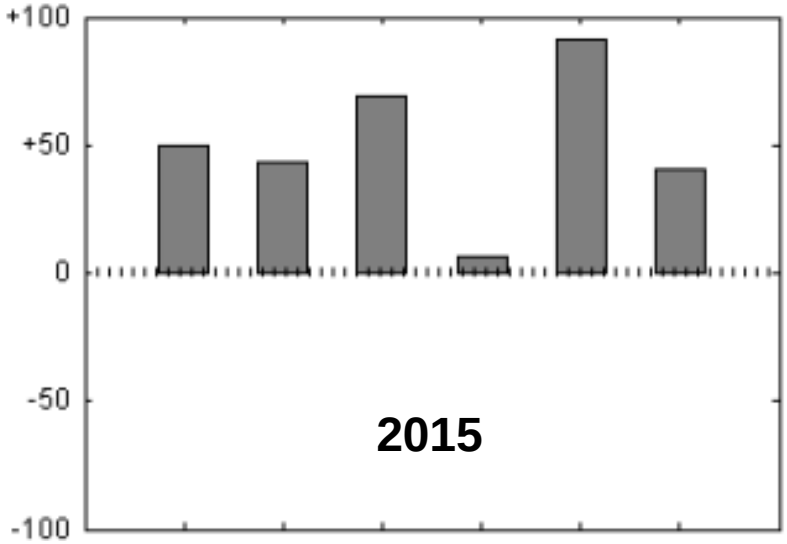
Steffen Malkowsky

Siyu Tan

Suggestion 10 – Team work

Course evaluation: ECQ





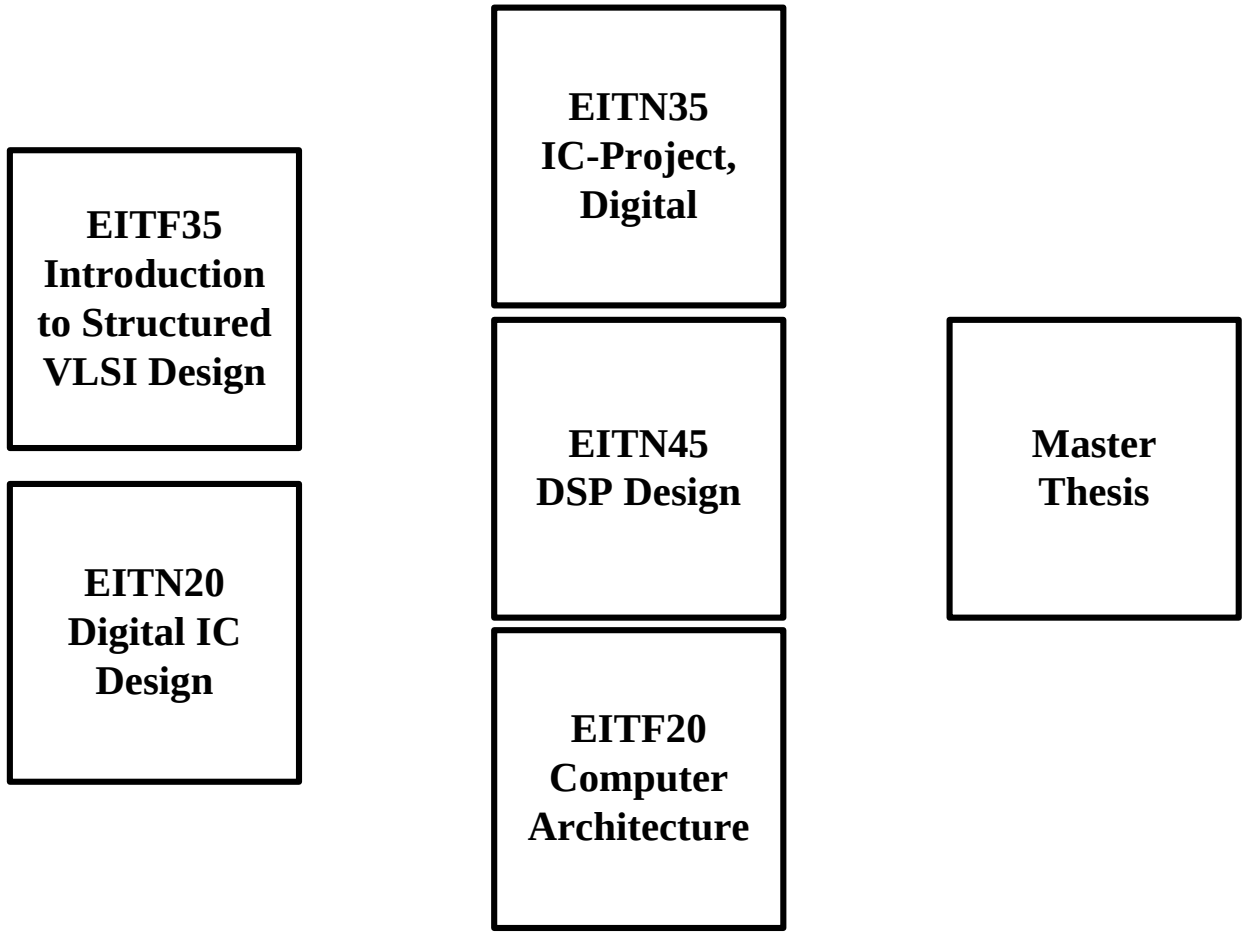
Outline

- Conclusion
- **Next Step**

This Course

- TA assistant will be reduced at the suggested deadline of project 4
- Need self problem-solving capability for project 5

Digital Path



The team – Digital Asic Group

Prof.
Viktor
Öwall

Assoc. Prof.
Erik
Larsson

Assoc. Prof.
Joachim
Rodrigues

Assoc. Prof.
Liang
Liu

Ph.D. Stud.
Mojtaba
Mahdavi

Ph.D. Stud.
Farrokh
Ghani Zadegan

Ph.D. Stud.
Rakesh
Gangarajaiah

Ph.D. Stud.
Yangxurui
Liu

Ph.D. Stud.
Steffen
Malkowsky

Ph.D. Stud.
Babak
Mohammadi

Ph.D. Stud.
Hemanth
Prabhu

Ph.D. Stud.
Dimitar
Nikolov

Ph.D. Stud.
Michal
Stala

Digital Asic Group – Projects, massive MIMO

imec

KATHOLIEKE UNIVERSITEIT LEUVEN

SONY

ERICSSON

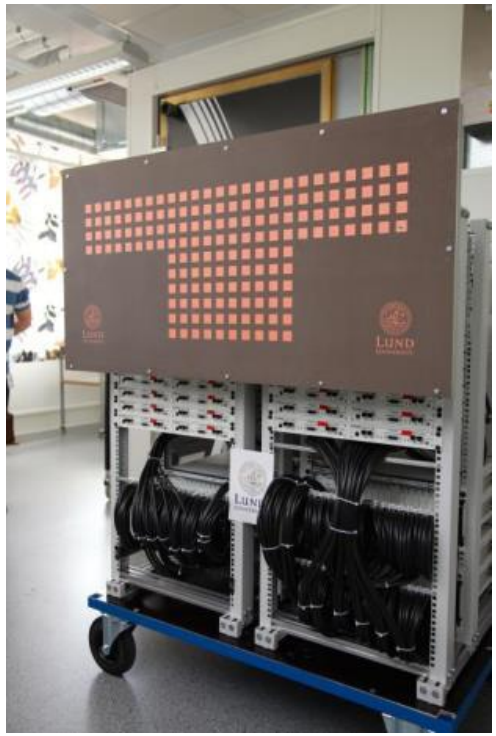
Linköpings universitet

Infineon

TECHNIKON

NATIONAL INSTRUMENTS

intel



100X

Digital Asic Group – project

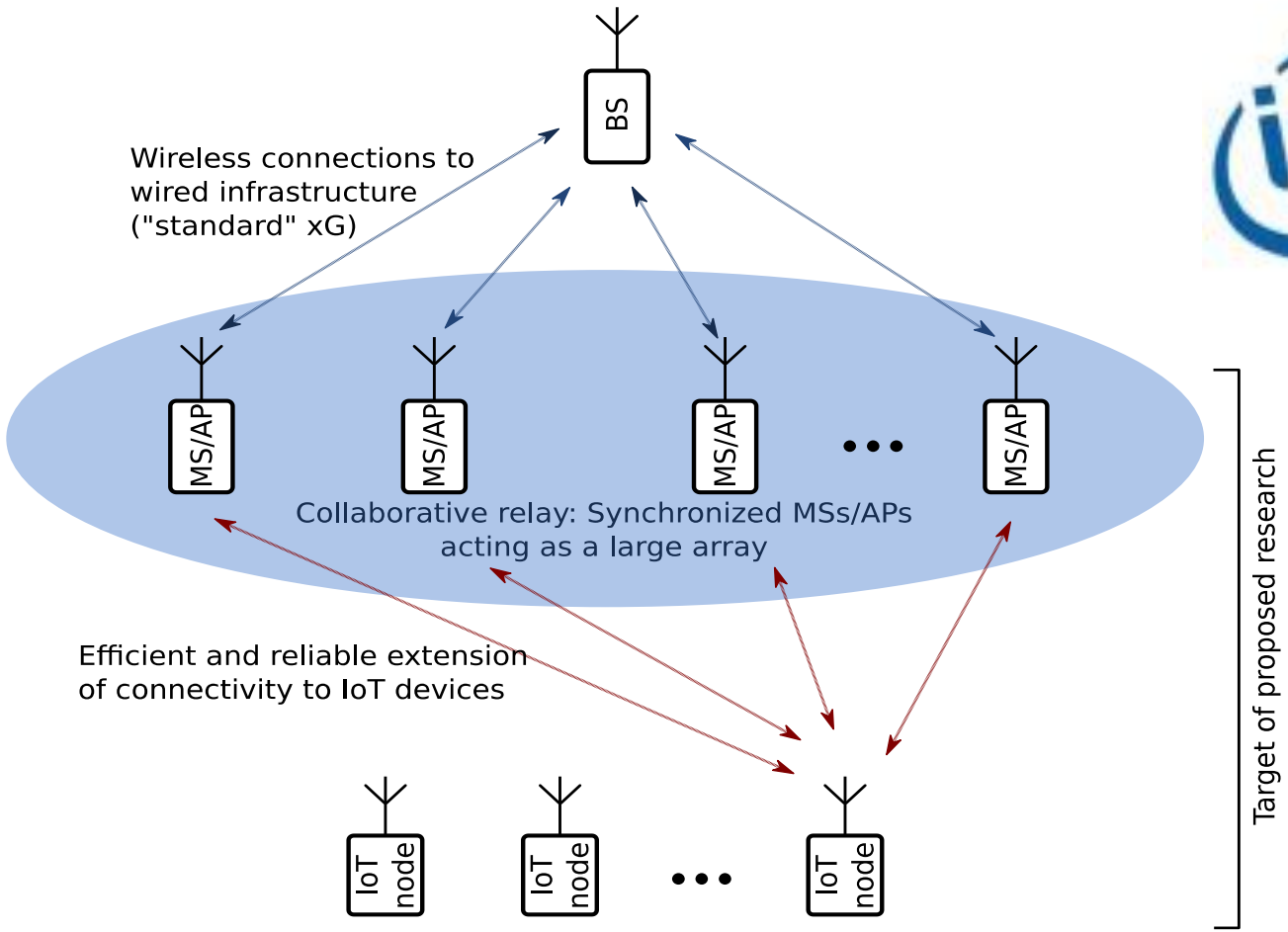
❑DARE: Digital Assistant Radio Evolution

- Everything now is digitalized
- Analog interface is still needed
- We are trying to digitalize the interface as much as possible

	Price
16-bit ADC	185$
10-bit ADC	50$

Digital Asic Group – Projects, beyond 5G

❑Coordination in Distributed Multi-User High-Performance Dense Networks



Digital Asic Group - Projects

- **Internet of Things: ultra-low power processor**

Digital Asic Group - Projects

- **Autonomous Systems, indoor localization**

http://www.automotive-fleet.com/video/detail/2016/09/volvo-produces-first-drive-me-project-car.aspx?refresh=true

Digital Asic Group - Projects

- **Digitizing system for Physics**

EUROPEAN SPALLATION SOURCE

Division of Chemical Physics

LUND UNIVERSITY

Atomic Physics

FACULTY OF ENGINEERING, LTH

Particle Physics

Nuclear Physics

Possible master thesis

AXIS COMMUNICATIONS

ARM

Stefan.Lundberg@axis.com

ERICSSON

charlotte.skold@ericsson.com

Start-Ups

Good Luck !