EITF35: Introduction to Structured VLSI Design

Part 7.1.1: Wrap Up

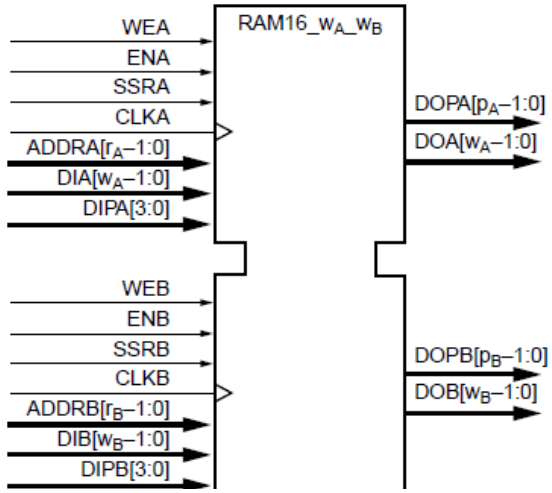
Liang Liu
liang.liu@eit.lth.se

Outline

- Conclusion
- Next Step

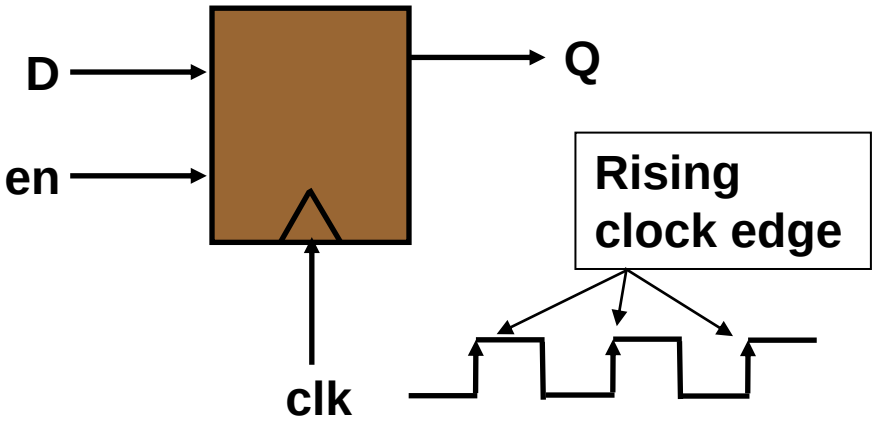
What we have learned

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

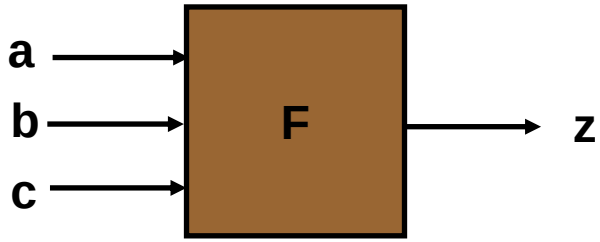


memory

Register

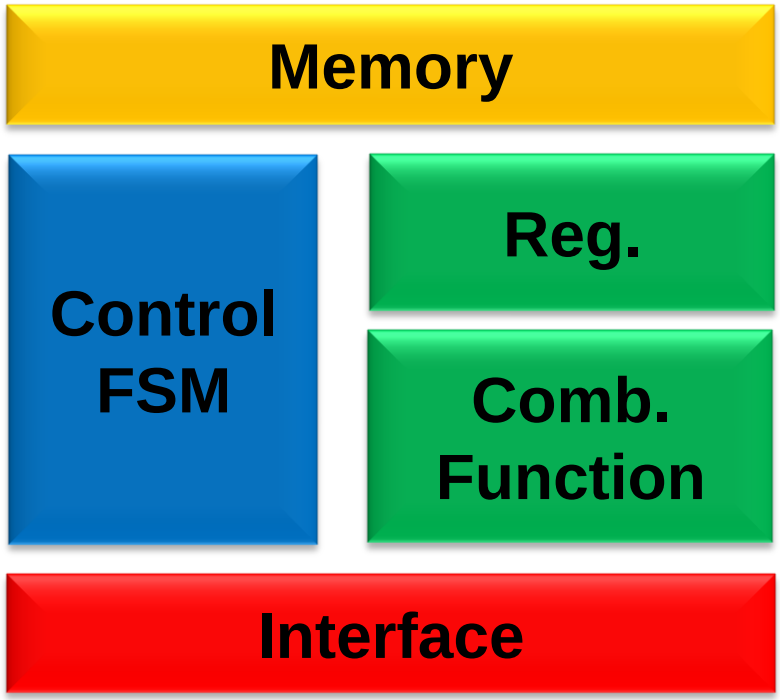


Combinational Logic



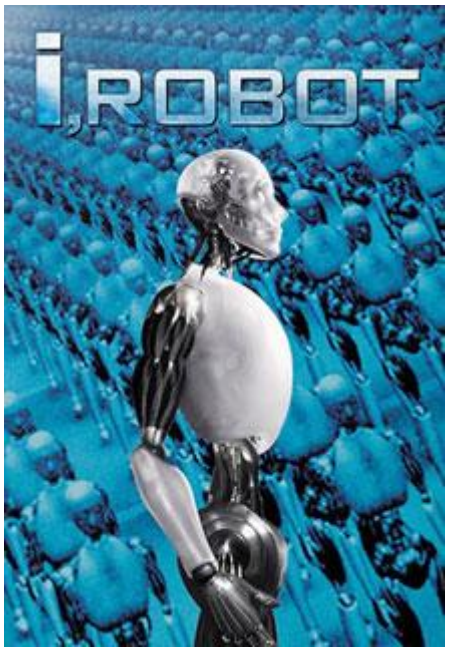
What, How, When

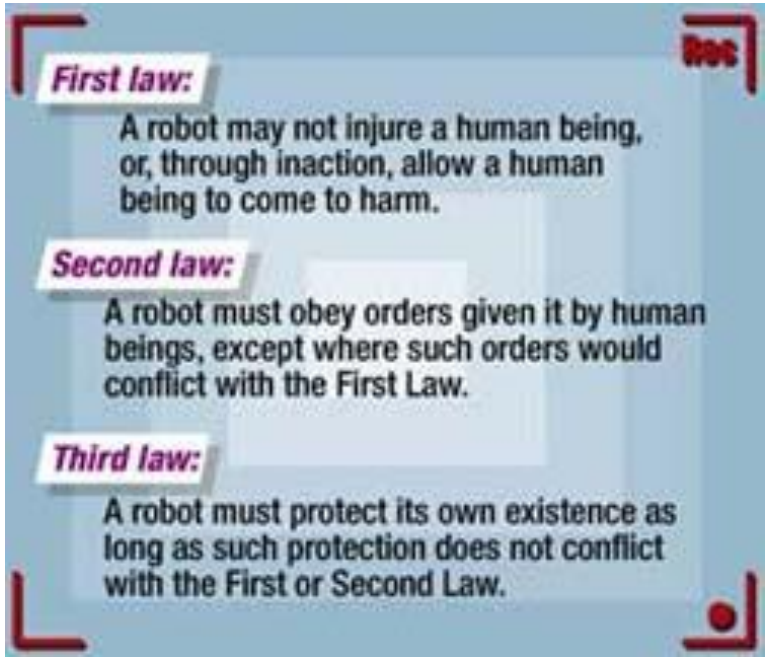
What we have learned



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

VLSI ‘Laws’





Workable hardware meeting the requirement

Suggestion 1 – Design Flow

Specification

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

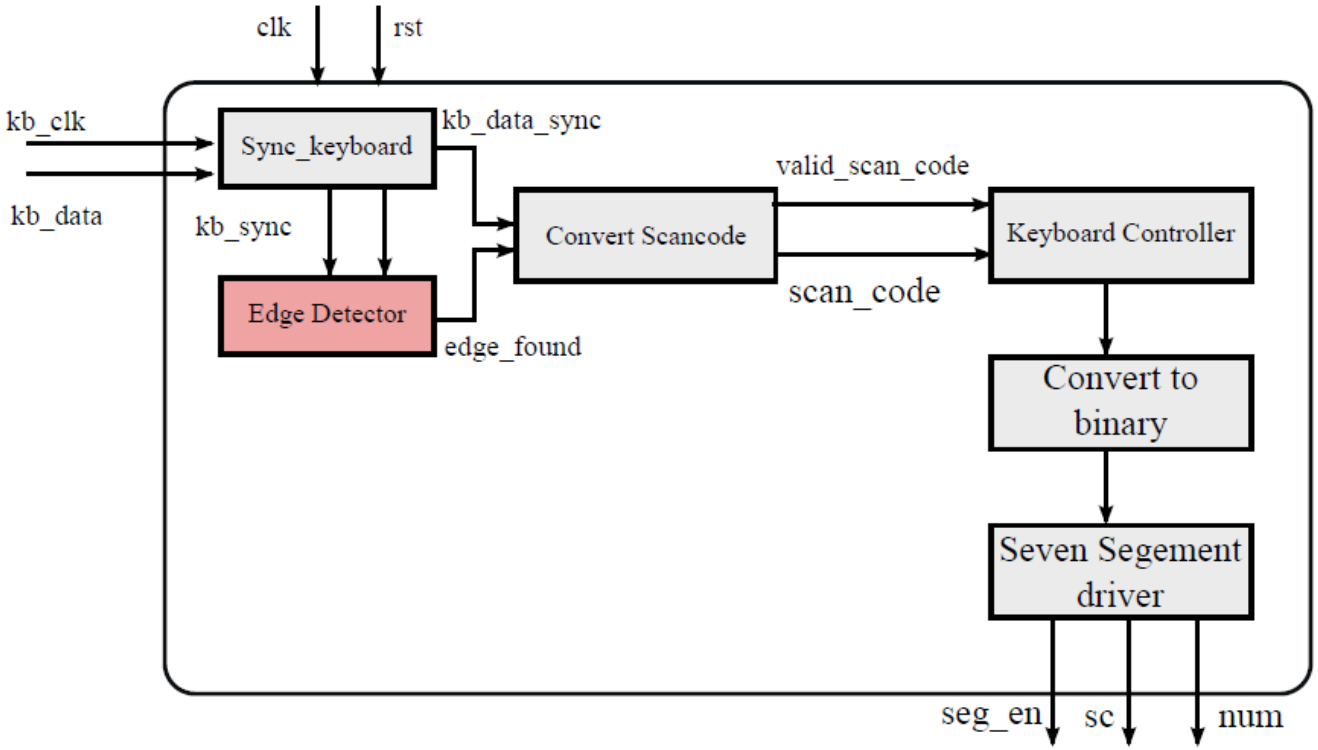
Rem or Mod?

Suggestion 1 – Design Flow

Specification

Block Diagram

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



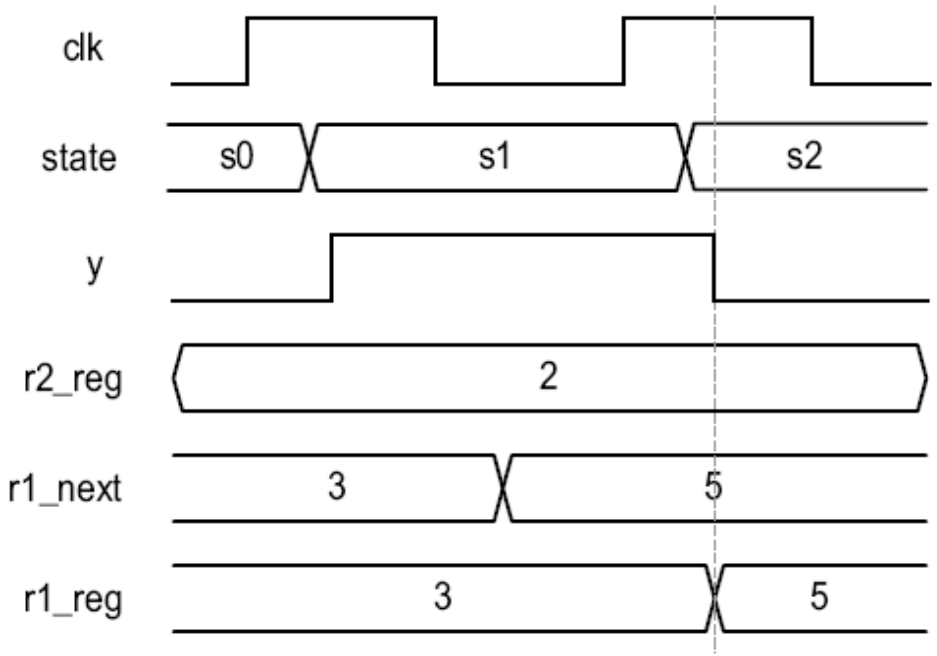
Suggestion 1 – Design Flow

Specification

Block Diagram

Timing Diagram

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



(c) Timing diagram

Suggestion 1 – Design Flow

- Specification
- Block Diagram
- Timing Diagram
- VHDL Coding
- Implementation

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

You are a architecture designer

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

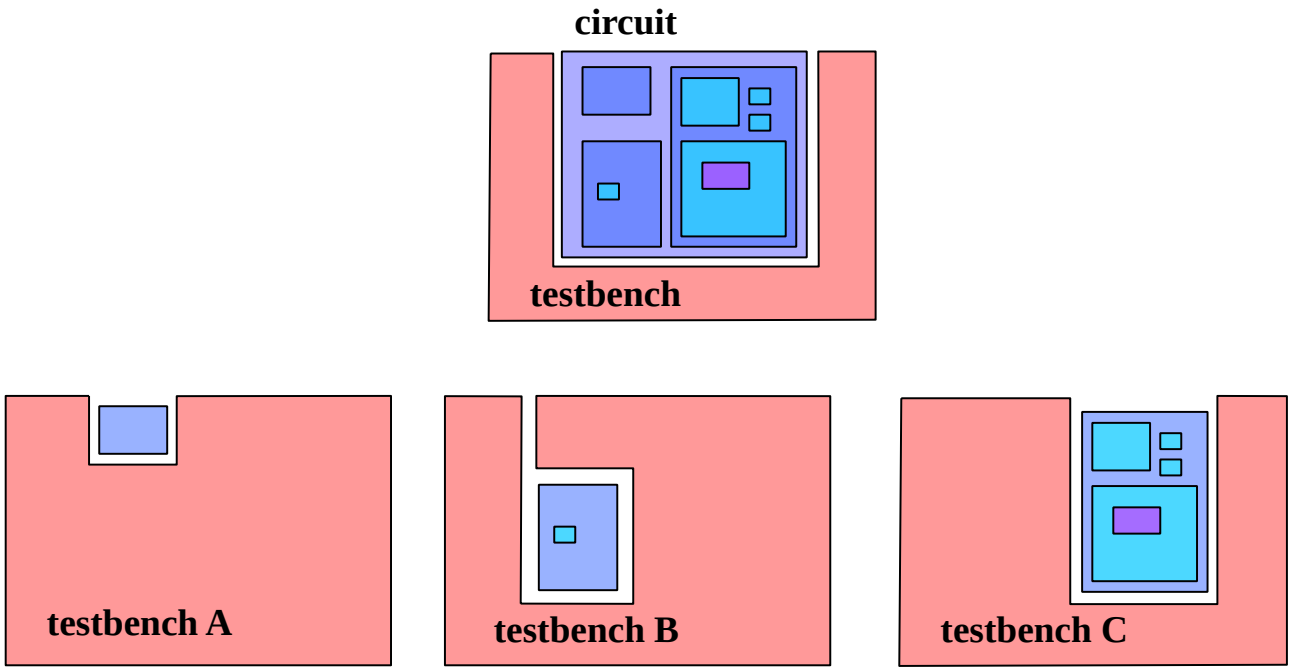
The cost of fixing a bug grows exponentially with the stage!!!

Suggestion 3 – Debug

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

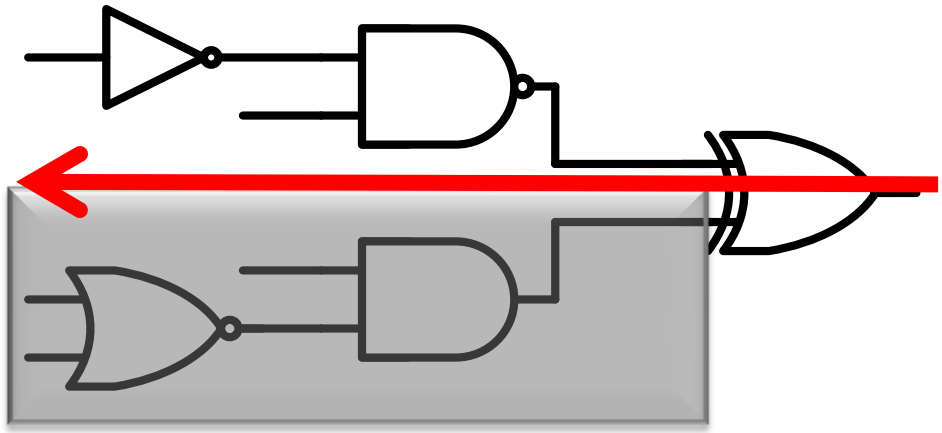
Suggestion 3 – Debug

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



Suggestion 3 – Debug

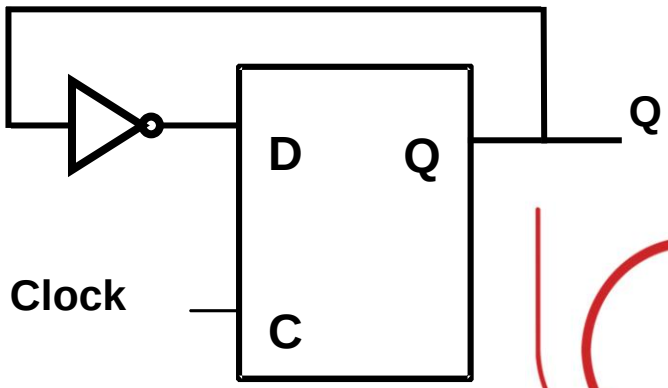
- **There MUST be an error**
  - Small & simple
- **Locate the error**
  - Divide and conquer
  - Error isolation
  - Trace back
  - Special stimuli

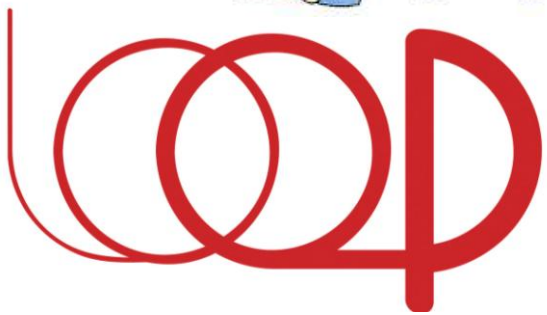


Suggestion 4 – Synthesizable VHDL

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

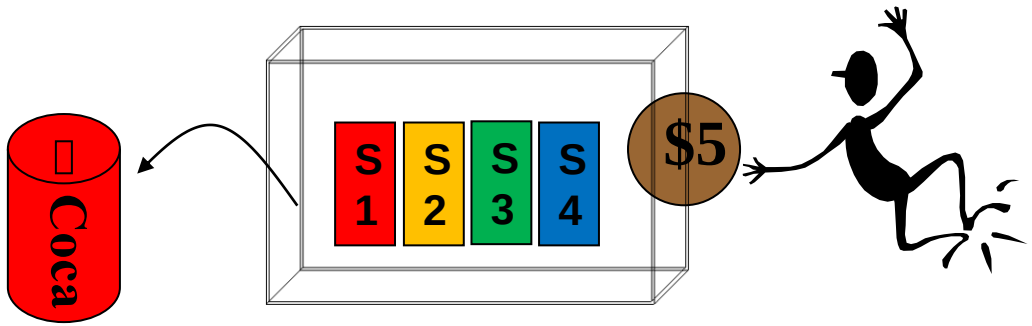
```
process (Clk, D)
begin
if rising_edge(Clk) then
Q <= D;
end if;
end process;
```

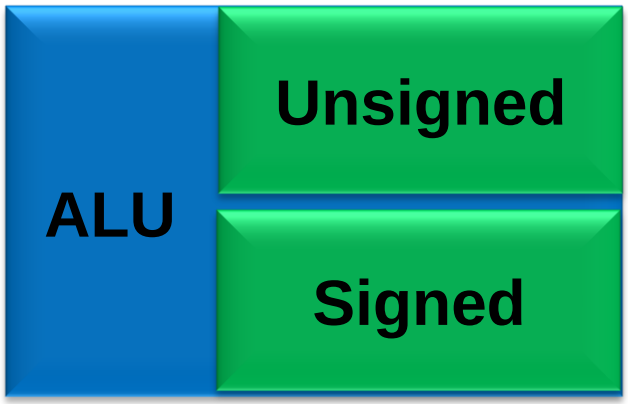




Suggestion 5 – Hierarchical Design

- **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)
```

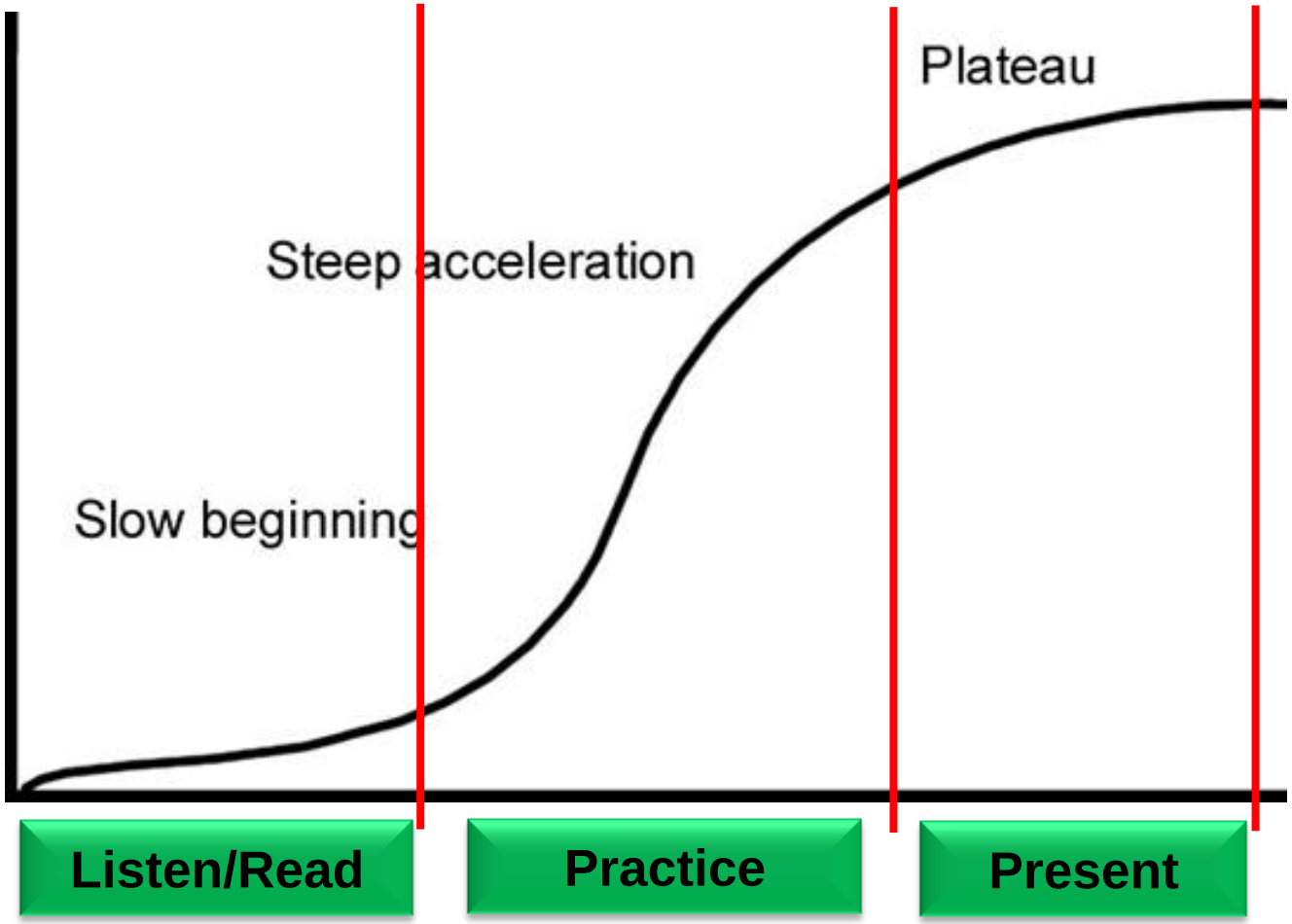
```
x = (x >> 6) + (x & 0x3f);
x = (x >> 4) + (x & 0xf);
x = (x >> 2) + (x & 0x3);
x = (x >> 2) + (x & 0x3);
x = (x >> 2) + (x & 0x3);
if (x == 3) x = 0;
```

	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



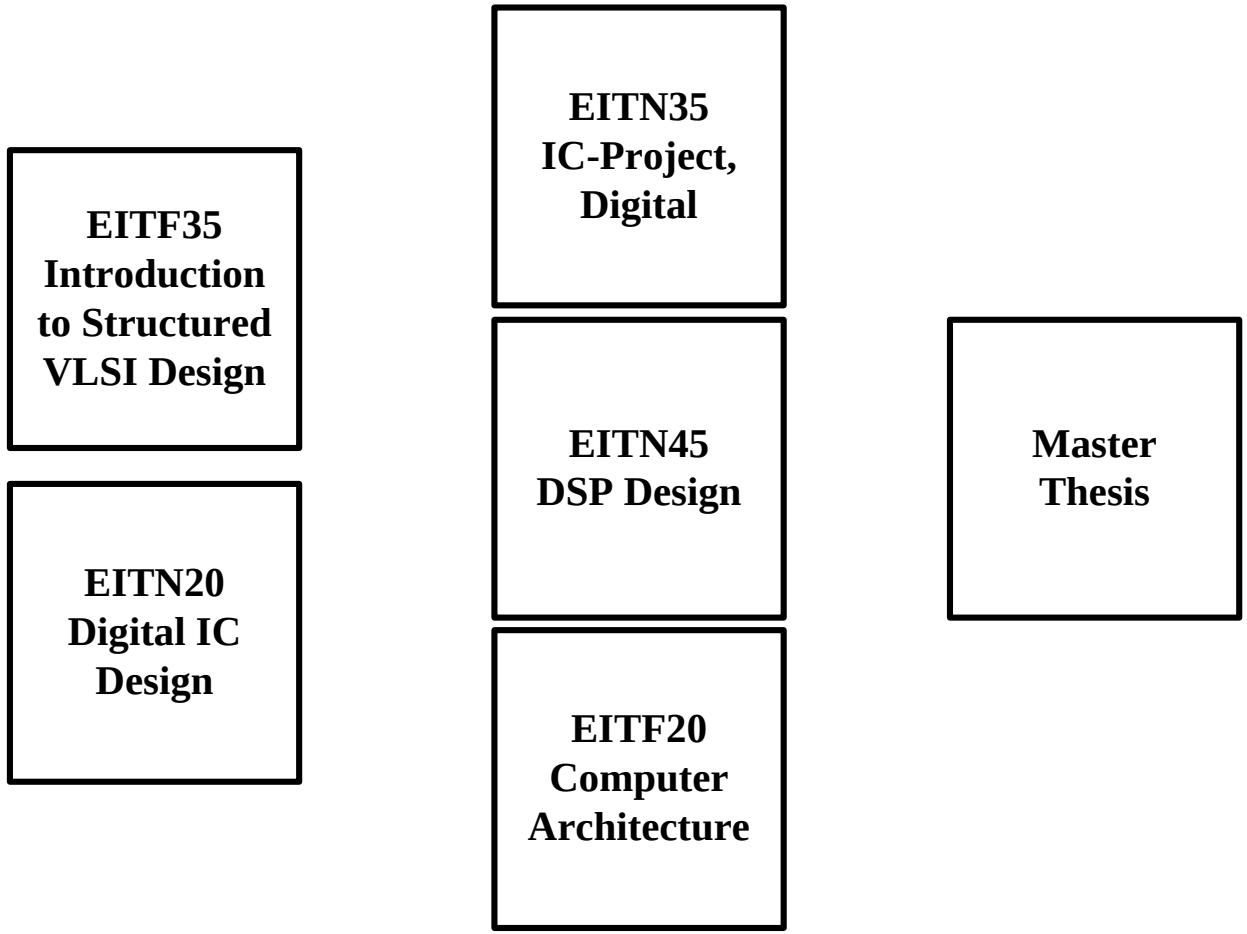
Outline

- Conclusion
- **Next Step**

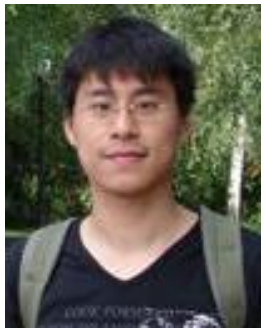
This Course

- TA assistant ends at 18th Oct. (deadline of project 4)
- Need self problem-solving capability for project 5

Digital Path



Digital Path



Chenxin Zhang

Intro. VLSI (mouse control)

IC Project (MIPS Processor)

Master Thesis (Multi-core MIPS)

PHD (Our own Processor)

Digital ASIC Group – The Team

Prof.
Peter
Nilsson

Prof.
Viktor
Öwall

Assoc. Prof.
Erik
Larsson

Assist. Prof.
Joachim
Rodrigues

Post doc
Liang
Liu

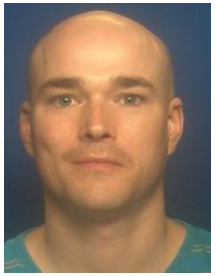
Ph.D. Stud.
Oskar
Andersson

Ph.D. Stud.
Anil
Dey

Ph.D. Stud.
Isael
Diaz

Ph.D. Stud.
Rakesh
Gangarajaiah

Lic.
Erik
Hertz



Ph.D. Stud.
Steffen
Malkowsky

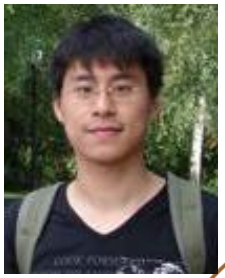
Ph.D. Stud.
Reza
Meraji

Ph.D. Stud.
Babak
Mohammadi

Ph.D. Stud.
Hemanth
Prabhu

Ph.D. Stud.
Yasser
Sherazi

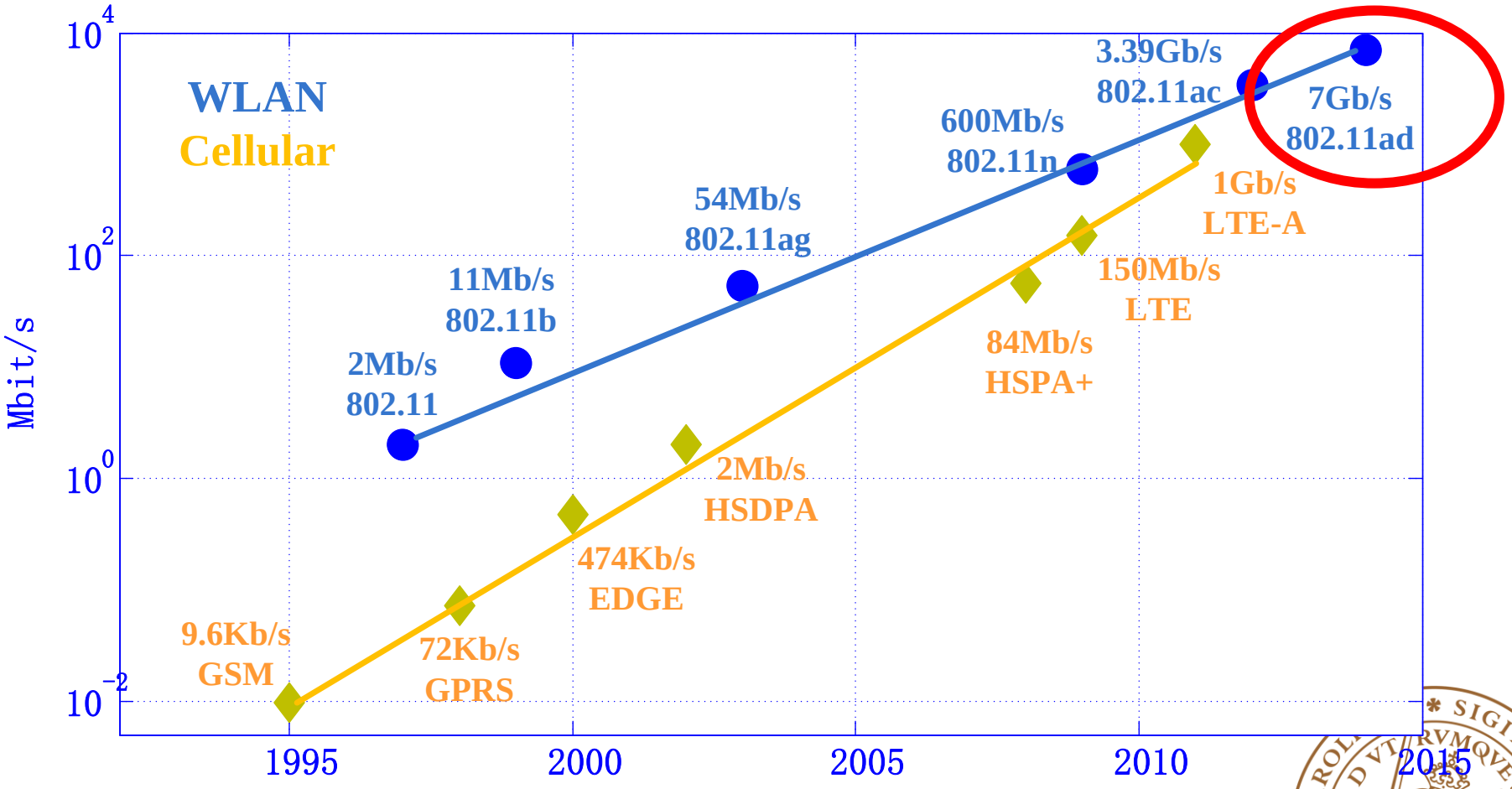
Ph.D. Stud.
Michal
Stala



Ph.D. Stud.
Chenxin
Zhang

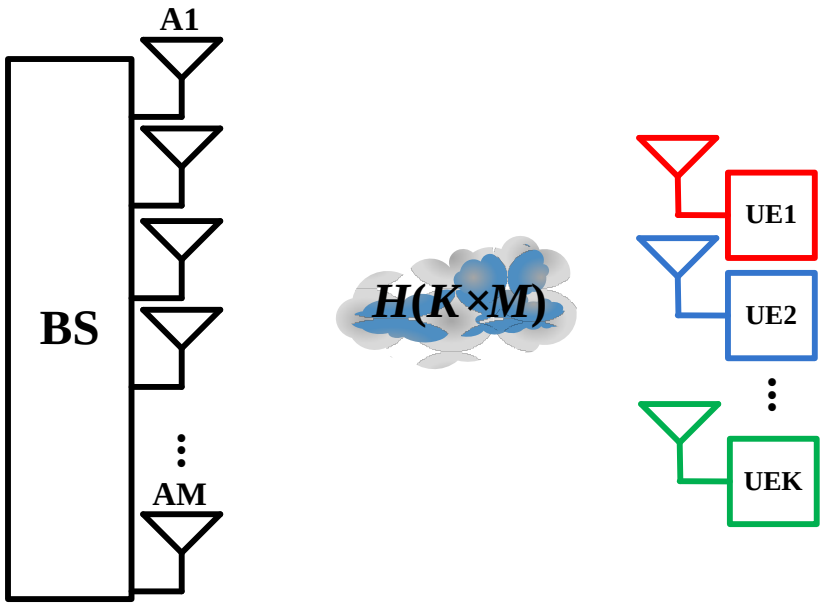
Digital Asic Group - Projects

- Massive MIMO: solution beyond LTE-A (4G)



Digital Asic Group - Projects

❑Massive MIMO: solution beyond LTE-A (4G)



- We are looking for **100 antennas** **at base station!**
- We serve **10-20 users** concurrently
- Reduce transmitting power or increase throughput

Digital Asic Group - Projects

❑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

❑ Adaptive Processing with Smart Circuit

- Changing environment & application
- Processor adaptive to requirement
- Energy efficient with satisfied QoS

❑ *Changing environment*

❑ *Various applications*

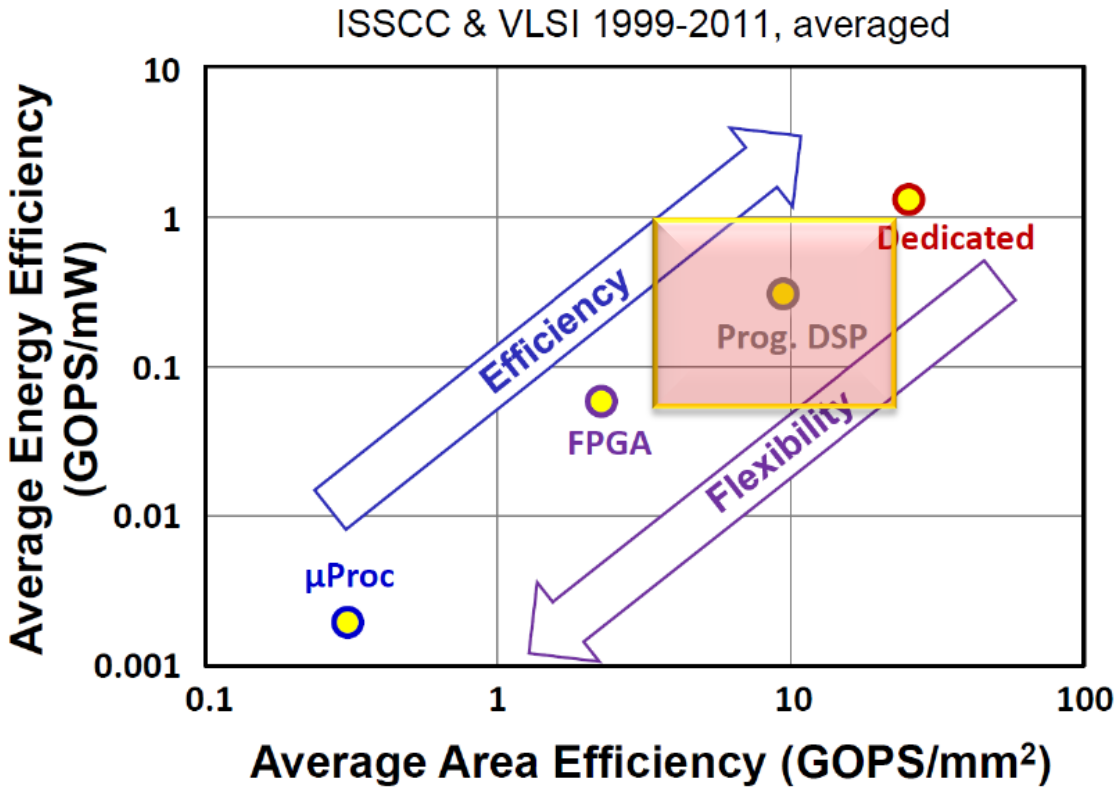
Smart Processor

❑ *Low Power Consumption*

Digital Asic Group - Projects

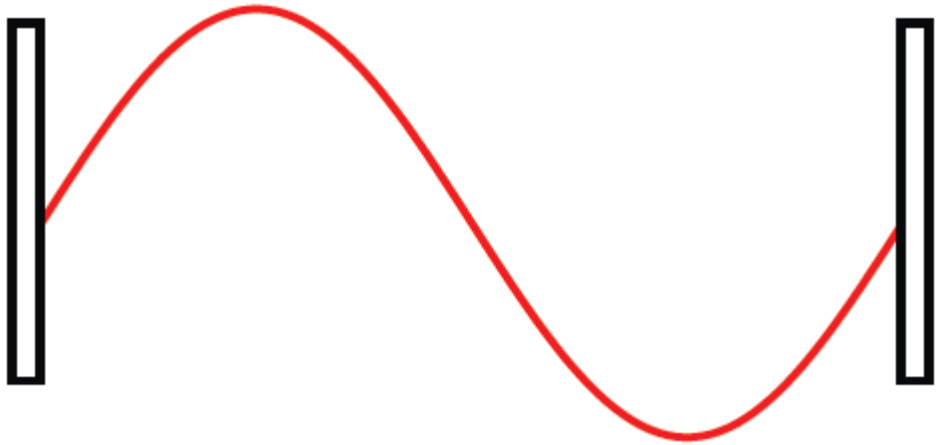
❑HiPEC: High Performance Embedded Computing

- Flexible to support multi-standard, multi-version, multi-mode
- Area & energy efficient to be adopted in portable device
- Application-specific processor (vector processor)



Digital Asic Group - Projects

- **Industry**
  - Ericsson, Axis, Sony Mobile…
- **Other Department**
  - Chemical Physics: digital cavity
  - Measure 1GHz signal with accuracy of 1Hz



Good Luck !!!