

Introduction

—Computer Organization—

Lionel Morel

Computer Science and Information Technologies - INSA Lyon

Fall-Winter 2025-26

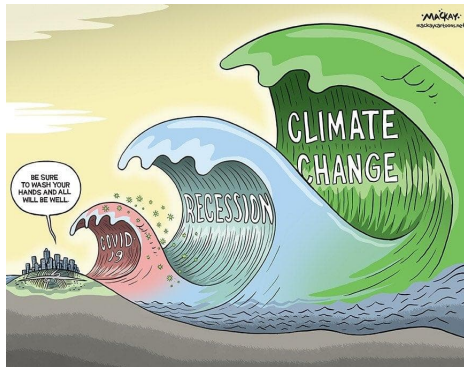
Lecturer - Lionel Morel¹ (lionel.morel@insa-lyon.fr)

- ▶ **96-2001**: BS and MSc in Computer Science, UJF-Grenoble
- ▶ **2001-05**: PhD INPGrenoble on Programming Languages for Safety-Critical Systems
- ▶ **2005-07**: Postdoc in Turku, Finland & Rennes, Brittany
- ▶ **2007-17**: Assoc. Professor, INSA Lyon, CS Dep (IF) - CITI Lab
- ▶ **2017-20**: Researcher at CEA-Grenoble
- ▶ **2020-...**: Assoc. Professor, INSA Lyon, CS Dep (IF) - CITI Lab
 - ▶ Teaching: Computer Architecture, Operating Systems, Compiler Construction, Environmental and Societal Challenges of Digital Technology
 - ▶ Research: Can we keep digital systems when living within Planetary Limits? What should they look like?
- ▶ Amateur musician and Professional Mountain Leader (part time)

¹<https://lionel.morel.ouvaton.org/>

Current Research and Teaching Interests

I know you heard, but we've set ourselves into serious trouble:
Human-Induced Climate Change, 6th Extinction, etc.



Current Research and Teaching Interests

Some (observed) solutions:

- ▶ “Don’t Look Up”, aka “Business as Usual”
- ▶ Abandon Ship (eg Grothendiek, students at Agro-Paris-Tech, and many others)
- ▶ Tech will save us (and the world) - “Techno-béatitude” (Musk, Electric Car, GDP/CO2 decoupling believers, etc)
- ▶ What if we stopped building Technology for itself and be more adult - “Techno-critique” - “Techno-réalisme”

Current Research (and Teaching) Interests

- ▶ Apply this idea to our field: operating systems, programming languages (and compilers a little bit)
- 1 What would mean frugality for computing systems: a hw+sw combination that would not last 3 years (average) but 10? 50? 100?
- 2 We are used to propose technology that go and impact society (and is actually imposed on it, through law, commercials, influences, etc). What about asking the society what it really needs ? What are “good” usages? What do real people need?

That's the sort of questions we try to answer in our group **Phenix @ CITI²**

²<https://phenix.citi-lab.fr/>

My take-away message

- ▶ Let's prepare for a rough ride
- ▶ Tech will NOT save us: it might be a part of the solution, but the main task is organizing society to accomodate for change and change the direction
- ▶ Let's build something new, not relying (specially) on tech
- ▶ Let's build something together (individualism is part of the problem)
- ▶ Let's enjoy this!

If any of this is of interest to you, please come and talk!

Context: Architecture - Operating System - Networks

▶ **3IF - Semester 1**

- ▶ **AC** - Architecture des Circuits / Digital Circuits (*Guillaume Beslon*)
- ▶ **AO** - Architecture des Ordinateurs / Computer Organization (*Lionel Morel*)
- ▶ **PRC** - Programmation C / C Programming (*Frédéric Prost*)

▶ **3IF - Semester 2**

- ▶ **SYS** - Systèmes d'Exploitation / Operating Systems (*Guillaume Salagnac*)
- ▶ **RE2** - Bases Techniques pour les réseaux / Networks 1 (*Frédérique Biennier*)

▶ **4IF - Semester 1**

- ▶ **SERE** Programmation Réseaux / Distributed Programming (*Frédéric Prost*)

▶ **4IF - Semester 2**

- ▶ **SERE** - Sécurité Réseau / Network Security (*Lionel Brunie*)
- ▶ **COMP** PLD Compilateur / Compiler Design (*Florent de Dinechin*)

AC, AO, SYS: Objectives

- ▶ SYS: “Basic understanding of the fundamental concepts and issues in the topic of operating systems”
- ▶ AC: “Gain theoretical and practical understanding of the principles and mechanisms that govern digital circuits, from basic logic gates up to a simple microprocessor. ”
- ▶ AO: “Understand:
 - ▶ the design and working principles of a modern computer (processor, memory hierarchy, peripherals)
 - ▶ the lower levels of the software stack: assembly language, application binary interface, interruptions, input/output drivers, compiler

AC - Content (reminder)

- ▶ Boolean logic
- ▶ Combinatorial Circuits
- ▶ Registers and Memories
- ▶ Finite-State Machines
- ▶ Sequential Circuits
- ▶ Algorithmic State Machines ...
- ▶ And their implementation with registers and gates

AO - Content of the lectures

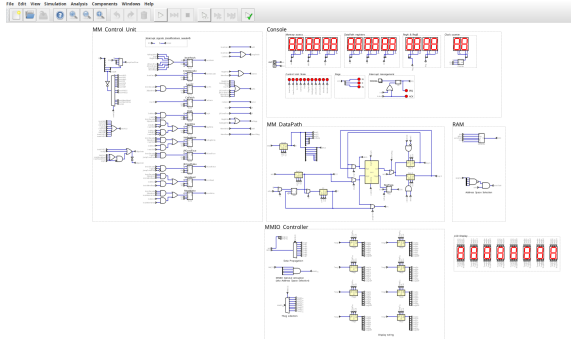
- ▶ Von Neuman Machine, Processor from the (low-level) programming perspective, assembly language
- ▶ Von Neuman Cycle, Execution perspective, Link between assembly language and processor's internals
- ▶ Input/Output, Interrupts, Memory-Mapped IO
- ▶ Function Calls, calling sequence, call/ret, parameter passing, recursion, return values

AO - Content of the Labs

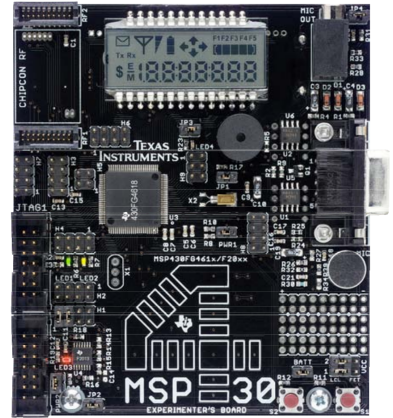
- ▶ Assembler programming “on paper”, simple algorithms (if-then-else, loops)
- ▶ Assembler programming on machine, assembly/disassembly, instruction encoding
- ▶ Memory-mapped IO
- ▶ Function calls (register-based parameter passing)
- ▶ Function calls (stack-based parameter passing)

AO - Content of the Labs

μ -Machine implem on Digital



Assembly programming on **msp430**



Practical matters - Agenda

Week	Lecture	Lab
38	AC-1: Information Coding	
39	AC-2: Boolean Logic	AC-1 (2hrs): Binary, number representations
40		
41	AC-3: Sequential Circuits & Memories	AC-2 (2hrs): Boolean Algebra
42		
43	AC-4: Finite State Machines	
44	Autumn break	
45	AC-5: Abstract State Machines	AC-3 (4hrs): Simple Combinatorial and Sequential Circuits
46	AC-6	AC-4 (4hrs): More Complex Sequential Circuits
47	AO-1: Von Neumann (ASM programming)	AC-5 (4hrs): Your first Von Neumann Machine!
48	AO-2: Von Neumann (ASM programming + Internals)	AO-1 (2hrs): Assembly Programming
49	AO-3: Von Neumann (Internals)	
50	AO-4: Inputs/Outputs	AO-2 (4hrs) : CPU design
51	AO-5: I/O & Function Calls	AO-3 (2hrs): CPU design
	AC Exam	
52	Christmas break	
1	Christmas break	
2	AO-6: Function Calls	AO-4 (4hrs): Assembly Programming / IO
3		
4		AO-5 (4hrs): Assembly Programming / IO
5	AO Exam	

Practical matters

Evaluation

AC: 2 ECTS

AO: 2 ECTS

Final grades:

- ▶ 1 exam for **AC**: week 51, **Thursday December 18th**, 08:00-10:00
(1h30 + 1/3 — 1 hand-written A4 sheet of notes (2 pages))
- ▶ 1 exam for **AO**: week 6, **Tuesday January 27th**, 10:00-12:00
(1h30 + 1/3 — 1 hand-written A4 sheet of notes (2 pages))

Others

- ▶ No ready-made solution to exercises provided
- ▶ But we answer all questions by email (or by appointment)
- ▶ Q/A Sessions **Every Monday** from 1 to 2 PM
Room 501.208.

IF/AO: People — `first.last@insa-lyon.fr`



Guillaume
Beslon



Romain
Bouarah



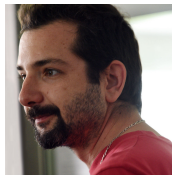
Florent
de Dinechin



Florent
de Dinechin



Jonathan
Rouzaud-Cornabas



Guillaume
Salagnac

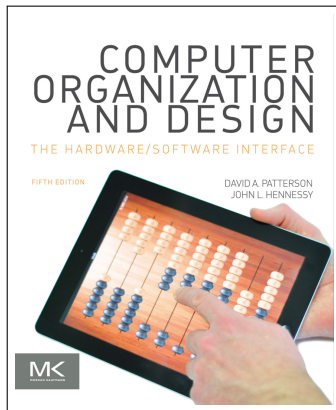


Lionel
Morel

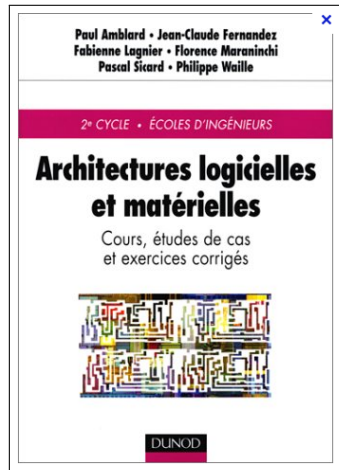
Readings



Architecture, Système,
Réseau 1
F. de Dinechin



Computer Organization
and Design
D. Patterson & J. Hennessy



Architecture Logicielle et
Matérielle
P. Amblard et al

All you need is ...

... on moodle ...



INSA
INSTITUT NATIONAL
DES SCIENCES
APPLIQUÉES
DE LYON

Accueil Tableau de bord Mes cours Aide ▾ Accès rapides ▾



Informatique / Informatique / Informatique / Informatique / IF-3

3IF - Architectures des ordinateurs

Cours Paramètres Participants Notes Rapports Plus ▾

▾ Organisation ↗

Course Info

- **Code:** IF-3-A0
- **ECTS:** 2.0
- **Lectures Hours:** 9hrs (6*1.5hrs)
- **Lab Hours:** 16hrs (2*2hrs + 3*4hrs)
- **Personal Work:** 25hrs
- **Language:** spoken French, lecture slides in English, labworks in French.
- **ECTS description:** FR/EN

Planning

Week	Lecture	Lab	MCQ
47	#1: Von Neumann (ASM programming)		
48	#2: Von Neumann (internals)	Lab 1 (2hrs): ASM programming	
49	#3: Inputs/Outputs		
50	#4: Inputs/Outputs	Lab 2 (4hrs): CPU design	
51	#5: Function Calls	Lab 4 (2hrs): CPU design	MCQ1
52	Christmas break		
1	Christmas break		
2	#6: Function Calls	Lab 4 (4hrs): ASM programming	
3			MCQ2
4		Lab 5 (4hrs): Interrupts	
5		Exam	

Personnel

Lecturer: Lionel Morel

Instructors: Guillaume Beslon, Lucas Chaloyard, Florent de Dinechin, Jonathan Rouzaud-Cornabas, Guillaume Salagnac,

<http://moodle2.insa-lyon.fr/course/view.php?id=1438>