Laurea Magistrale in Ingegneria Informatica

Architetture dei Sistemi di Elaborazione

ASE (02GOLOV)

Master's degree in Computer / Electronic Engineering

Computer Architectures

CAs (02LSEOQ, 02LSEOV, 02LSEXW)

Paolo BERNARDI – Ernesto SANCHEZ – Renato FERRRERO

Main responsible **ASE (AAA-GRA)**

- Paolo BERNARDI
- Department of Control and Computer Engineering
- DAUIN (4th floor).
- http://goo.gl/Zg5wE
- Tel. 011 090 7183
- e-mail paolo.bernardi@polito.it

Main responsible **ASE (GRB-ZZZ)**

- Ernesto SANCHEZ
- Department of Control and Computer Engineering
- DAUIN (4th floor).
- http://goo.gl/Zg5wE
- Tel. 011 090 7182
- e-mail ernesto.sanchez@polito.it

Main responsible **CAs**

- Renato FERRERO
- Department of Control and Computer Engineering
- DAUIN (3rd floor).
- http://goo.gl/Zg5wE
- Tel. 011 090 7177
- e-mail renato.ferrero@polito.it

Teaching assistants ASE (AAA-GRA)

- Francesco Angione
- Tel. 011 090 7091
- e-mail

francesco.angione@polito.it

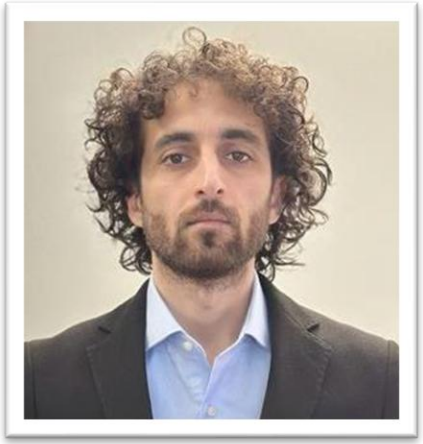
- Giorgio Insinga
- Tel. 011 090 7091
- e-mail

giorgio.insinga@polito.it

Teaching assistants ASE (GRB-ZZZ)

- Farnaghinejad Behnam
- Tel. 011 090 7091
- e-mail

behnam.farnaghinejad@polito.it



- Antonio Porsia
- Tel. 011 090 7091
- e-mail

antonio.porsia@polito.it

Teaching assistants CAs - Lectures

- Matteo Sonza Reorda
- Tel. 011 090 7055
- e-mail

matteo.sonzareorda@polito.it

- Marco Russo
- Tel. 011 090 7087
- e-mail

marco.russo@polito.it

Teaching assistants Cas - Laboratories

- Nicola Di Gruttola
- e-mail

nicola.digruttola@polito.it

- Gabriele Filipponi
- e-mail

gabriele.filipponi@polito.it

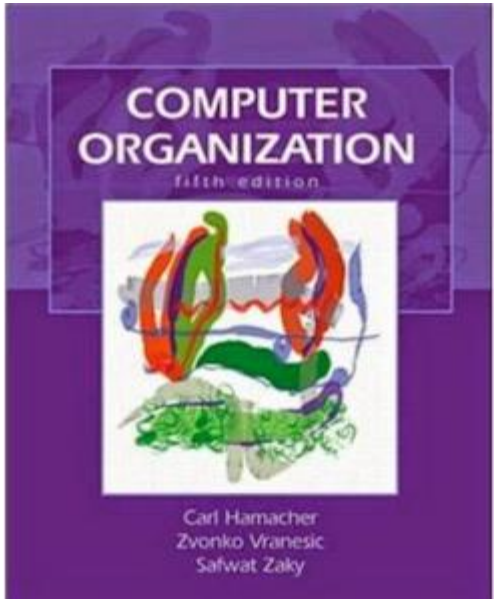
Goal of the course

- The course aims at providing the students with a basic knowledge about the architecture of modern microprocessor-based systems, with special emphasis on the microprocessor core
- The course is structured in two main blocks:
  1. Modern processor architecture
  2. ARM-based system
     a) ARM assembly language
     b) ARM Cortex-M3 board

Prerequisites

- The course is better followed if the student owns the basic knowledge about
  - The basic architecture of a computer system
  - The basic architecture and behavior of microprocessor-based systems
- Some programming knowledge is also assumed.

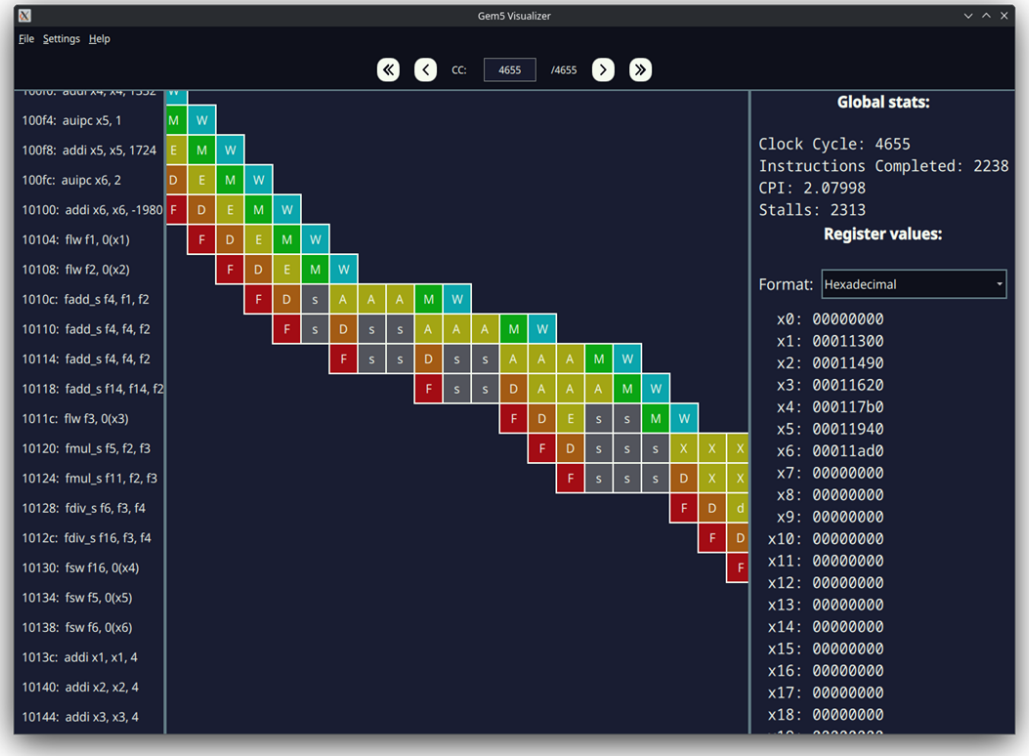
Computer Organisation: Fifth Edition
Carl Hamacher, Zvonko Vranesic, Safwat Zaky



Content

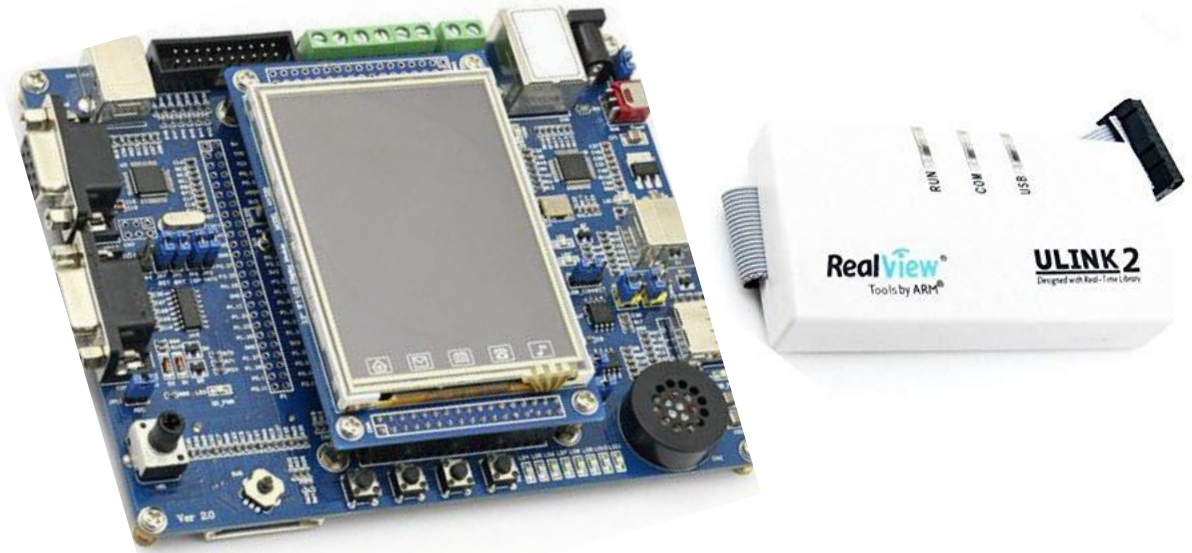
- Modern processor architecture
  - Introduction to computer design
  - RISC processors architecture and behavior
  - Superscalar processors
  - Disrupting architectures.

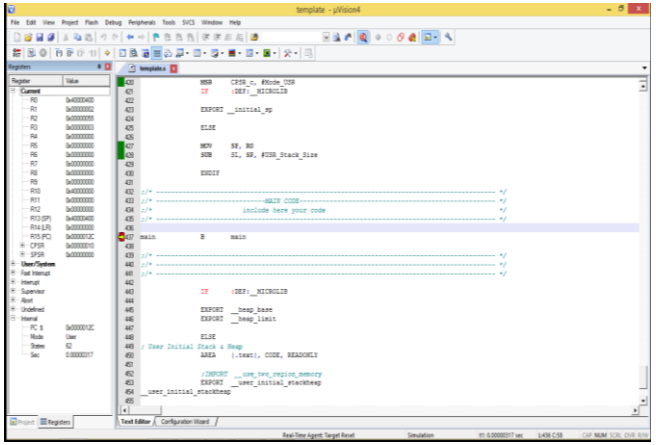
Gem5 & RISC-V



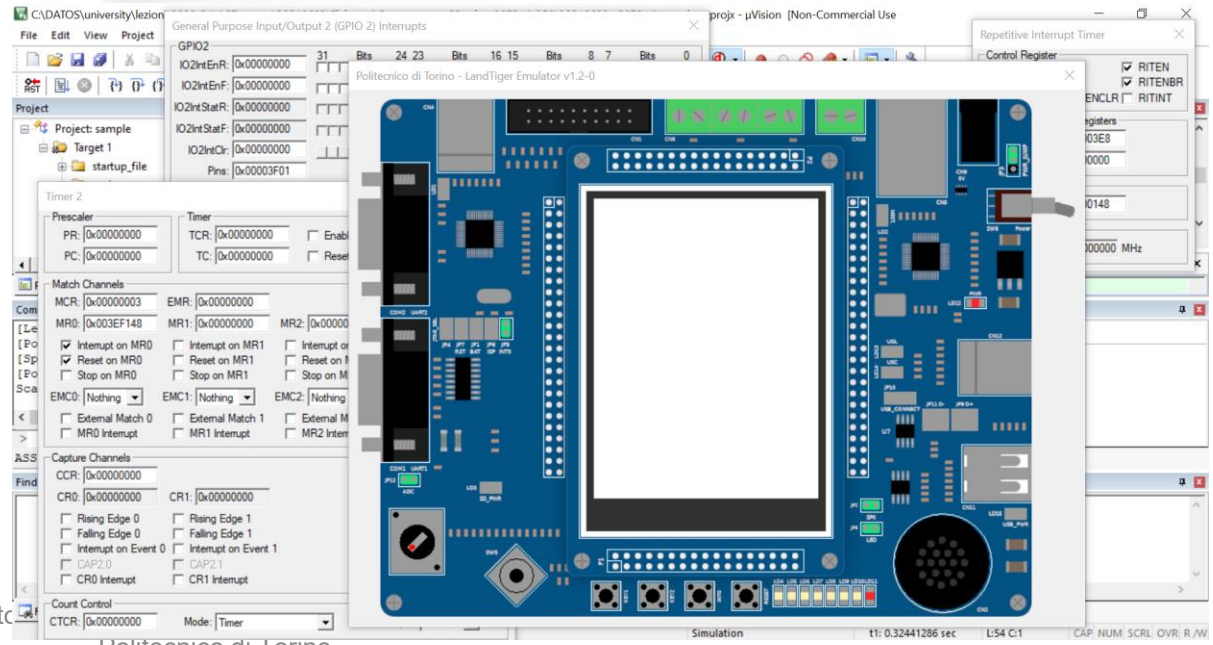
Content - 2

- ARM-based systems
  - Basic architecture
  - Assembly language
  - Development board
  - Emulation system.





Keil µVision



Lectures ASE (AAA-GRA)

LAB groups:

1. WEDNESDAY 11:30 – 13:00 → students from AAA - BBB
2. WEDNESDAY 13:00 – 14:30 → students from CCC - DDD

	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
08:30 – 10:00		**Lecture 3P**			
10:00 – 11:30	**Lecture 27**	**Lecture 3P**			
11:30 – 13:00	**Lecture 27**		**LABInf**		
13:00 – 14:30			**LABInf**		
14:30 – 16:00					
16:00 – 17:30					
17:30 – 19:00					

Lab starts at week 3

Lectures ASE (GRB-ZZZ)

LAB groups:

1. Wednesday 8:30 – 10:00 → students from COLELLA - PERSICI
2. Wednesday 10:00 – 11:30 → students from PINO - ZUNINO

	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
08:30 – 10:00		**Lecture 10A**	**LABInf**		
10:00 – 11:30	**Lecture 16**	**Lecture 10A**	**LABInf**		
11:30 – 13:00	**Lecture 16**				
13:00 – 14:30					
14:30 – 16:00					
16:00 – 17:30					
17:30 – 19:00					

Lab starts at week 3

Lectures COMPUTER ARCHITECTURES

LAB groups:

1. Wednesday 14:30 – 16:00 → students from A to K
2. Thursday 14:30 – 16:00 → students from L to Z

	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
08:30 – 10:00		**Lecture R3**			
10:00 – 11:30	**Lecture 5**	**Lecture R3**			
11:30 – 13:00	**Lecture 5**				
13:00 – 14:30					
14:30 – 16:00			**LABInf**	**LABInf**	
16:00 – 17:30					
17:30 – 19:00					

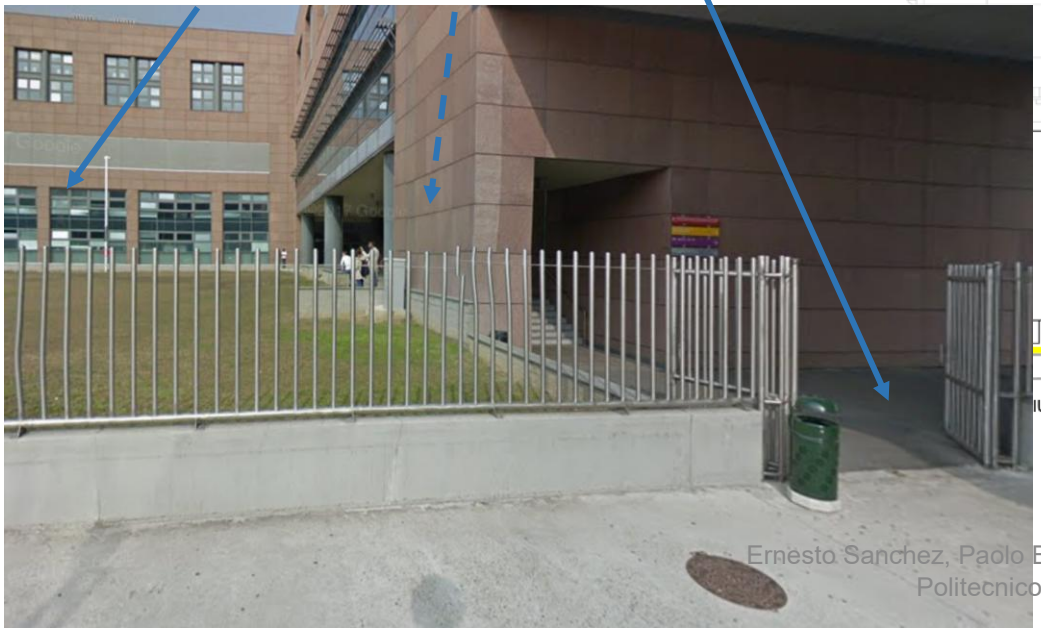
Lab starts at week 2

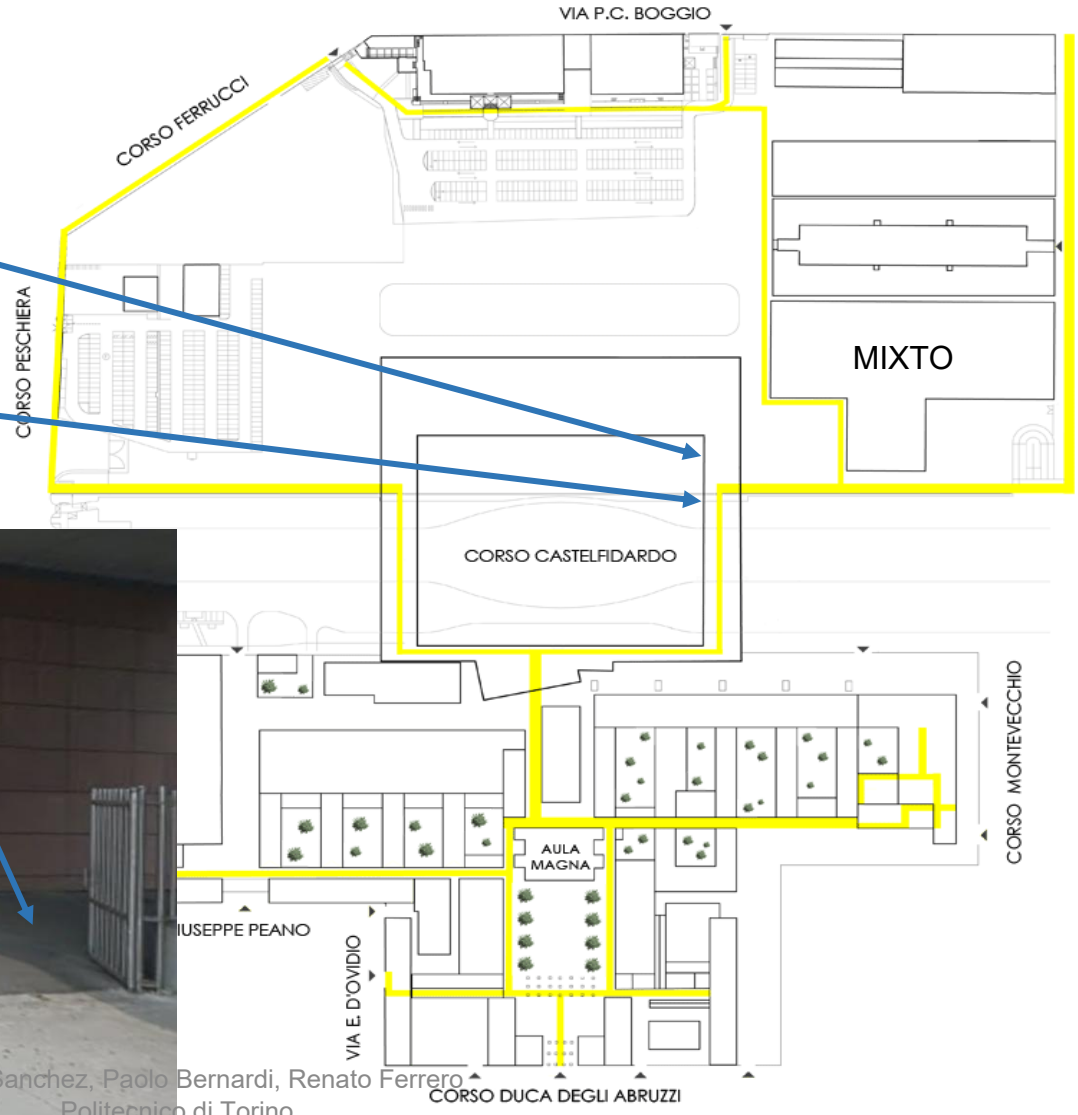
LABINF – Laboratorio Didattico di Informatica Avanzata
Location

LABINF:
1st floor "scavalco nord"

Accessible from:
Corso Catelfidardo 34C

GM Powertrain Europe





LABINF account

- During the first week, you will receive an email from LABINF with username and password
- If not, visit the LABINF and contact personnel to create your account before the first lab.

Labs

- Students will be guided to perform some practical experiences on:
- High level architectural simulators
  - gem5 – RISC-V
- ARM Development system and board
  - Keil µVision 5
  - Landtiger board based on a NXP ARM-based System-on-Chip
- There will be 12 lab weeks (we start at week 3)
  1. Regular tracks for the first **10** labs (**N = 10**)
  2. Two-phases extra-point project during last 2 weeks
- The student can upload the filled track and the solution by a strict deadline in the “elaborate – Homework” session of the teaching portal.

Support for remote training by a new, unique, very accurate EMULATOR integrated in the Keil debugger

Regular labs track example

Architetture dei Sistemi di Elaborazione [AA-LZ]	Delivery date: Tuesday 22 October 2019
Laboratory 1	Expected delivery of lab_01.zip including: - program_1.s - lab_01.pdf (fill and export this file to pdf)

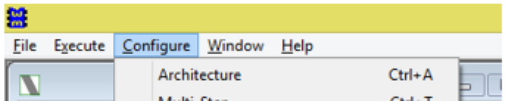
Please, configure the winMIPS64 processor architecture with the *Base Configuration* provided in the following:

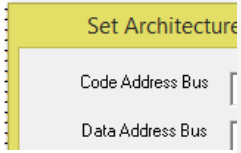
- *Integer ALU: 1 clock cycle*
- *Data memory: 1 clock cycle*
- *Branch delay slot: 1 clock cycle*
- Code address bus: 12
- Data address bus: 12
- Pipelined FP arithmetic unit (latency): 6 stages
- Pipelined multiplier unit (latency): 8 stages
- divider unit (latency): not pipelined unit, 28 clock cycles
- Forwarding optimization is disabled
- Branch prediction is disabled
- Branch delay slot optimization is disabled.

Use the Configure menu:

- remove the flags (where activating Enable options)
- Browse the Architecture menu →

Modify the defaults Arcl parameters (where neede





1) Exercise your assembly skills and learn by example about pipeline optimizations. To write an assembly program called **program_1.s (to be delivered)** for the *MIPS64* architecture and to execute it. The program has to search for the maximum integer number in a vector of 100 elements stored in memory; each element of the vector is 64-bit wise and contains signed integer values. The program saves the obtained value in a variable allocated in memory, called result.

   Identify and use the main components of the simulator:

   a. Running the *WinMIPS* simulator
      - Launch the graphic interface
      ...\winMIPS64\winmips64.exe

   b. Assembly and check your program:
      - Load the program from the `File→Open` menu (*CTRL-O*). In the case the of errors, you may use the following command in the command line to compile the program and check the errors:
      `...\winMIPS64\asm program_1.s`

   c. Run your program step by step (*F7*), identifying the whole processor behavior in the six simulator windows:
      Pipeline, **Code**, **Data**, **Register**, **Cycles** and **Statistics**

   d. Enable one at a time the optimization features that were initially disabled and collect statistics to fill the following table **(fill all required data in the table before exporting this file to pdf format to be delivered)**.

Table 1: **Program performance for different processor configurations**

	Number of clock cycles			
Program	No optimization	Forwarding	Branch Target Buffer	Delay Slot
`program_1`				

Extra-points

- All students delivering the solutions of **at least 9 of the 10 laboratories on time,** qualify to receive ***lab extra-points*** with a final project
- The extra-points project will be proposed during the last 2 labs
- Detailed rules
  - The last delivery date is **January 24, 2026**
  - It grants **up to 4 extra-points to be added to exam mark** (provided that the exam mark is greater or equal than 18)
  - Extra-points validity is 1 year (4 exam sessions, until September 2026).

Final project example

Computer Architectures 2018-19
02LSEOV 02LSEOQ [AA-LZ]

Delivery date:
Thursday 10/1/2019

Extra-points Project
Part 1

Expected delivery of **extrapoint_01.zip** must include:
- The zipped folder of your project
- A 4 minutes video (.mp4 or .avi) showing a running software debug session with significant peripheral windows shown.

Purpose of Part 1: to acquire full confidence in the usage of the KEIL software debug environment to emulate the behaviour of the LPC1768 and the LANDTIGER Board.

This part is evaluated to assign a maximum of 2 extra-points for qualified students taking the exam with vote >= 18

Start from the 16b_sample_BUTTON_LED_NVIC_PCON_TIMER project to develop the controller of a pedestrian crossing semaphore.

You are asked to write a program for the LandTiger Board that permits to reproduce the behaviour of a simple semaphore on a pedestrian crossing. An example is provided in figure 1. The crossing is regulated by
- 2 types of traffic lights:
  - 2-lights pedestrian traffic light (see figure 2)
  - 3-lights pedestrian traffic light (see figure 3)
- Pushbutton panels for pedestrian request see (figure 4)

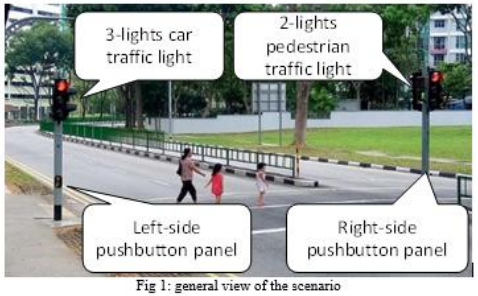


Fig 1: general view of the scenario



Computer Architectures
02LSEOV 02LSEOQ [AA-LZ]

Delivery date:
Sunday 20/1/2019

Extra-points Project
Part 2

Expected delivery of **extrapoint_02.zip** must include:
- The zipped folder of your project
- A 2-pages "application note" in pdf format: the application note is intended to briefly describe the structure of your project, with a synthetic explanation about the configuration of the used system components.

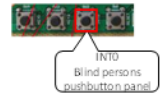
Purpose of Part 2: to acquire full confidence in the usage of the LANDTIGER Board.

This part is evaluated to assign a maximum of 2 extra-points for qualified students taking the exam with vote >= 18

Start from the extrapoint_01 project to implement an advanced version of the controller of the pedestrian crossing semaphore.

You are asked to write a project for the LandTiger Board that implements the following additional functionalities with respect to the basic behaviour already implemented.

A) Button INT0 is used to receive a crossing request from blind persons. This button fully complies with the current behaviour (please see the state diagram of the extra-points lab 1). The additional behaviour to be implemented is the following:
   1) As soon as INT0 is pressed, the loudspeaker emits a confirmation sound until the button is released.
   2) When the traffic light is green for pedestrians, the loudspeaker alternates 1s sound ON and 1 s OFF
   3) When the traffic light is flashing green for pedestrians, the loudspeaker alternates sound ON and OFF synchronously with the flashing frequency (according to previous specifications).
   4) When the traffic light for pedestrian is red, the sound is OFF.



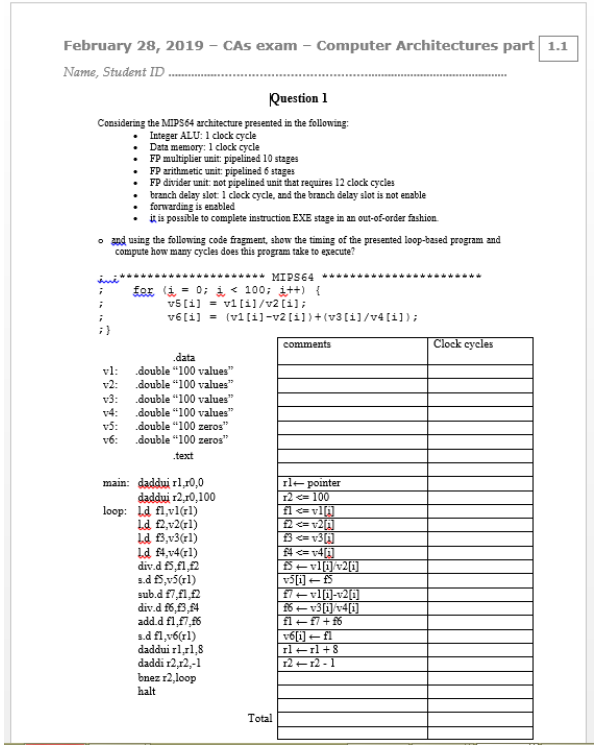
B) The Joystick is used to enter into a configuration mode of the traffic light system. This configuration can be entered only in the car=red / pedestrian=green state (please, see the state diagram below).
   1) When the joystick switch is pressed to the right, the traffic light system enters into the "maintenance" status by:
      i. Flashing red for pedestrians
      ii. Flashing yellow for cars
      iii. Flashing frequency is 1 s ON / 1 s OFF
      iv. The loudspeaker alternates ON/OFF sound with the same frequency of the lights
      v. The normal behaviour is restored when the Joystick switch is pressed to the left
   2) In this mode, it is also possible to use the potentiometer to regulate the loudness of the loudspeaker
      i. As soon as the potentiometer regulator is turned, the loudspeaker tone loudness is modified according to the regulator position

Final examination – in presence

- The final examination is done at the LABINF and it is composed of 2 parts:
  1. Processor architecture related exercises and questions
     - Exercises to be solved by hand
     - Open questions
     - No material can be used during this part
     - A sufficient mark (>=18) on this part is required to access the second
  2. ARM-based board programming exercise
     - To be taken at the computer (must pass a compile check)
     - Students will be allowed to use all the material provided in the course page in the polito site and their own projects
     - It will be corrected only if the first part is sufficient (>=18) and the project correctly compiles
- The whole exam lasts 2 hours.
- The written part mark is computed as the average of the two parts, provided that both of them are sufficient (>=18/30 each)
- Extra points rewarded by the final project are summed to the written mark
- Given that the exam returns marks up to 34, the 30 cum laude is given for marks higher than or equal to 33 (>32.5).

Exam example and evaluation flow-chart

Modern Architectures
(part 1 – 1h)



February 28, 2019 – CAs exam – Computer Architectures part 1.1

Name, Student ID .....

Question 1

Considering the MIPS64 architecture presented in the following:

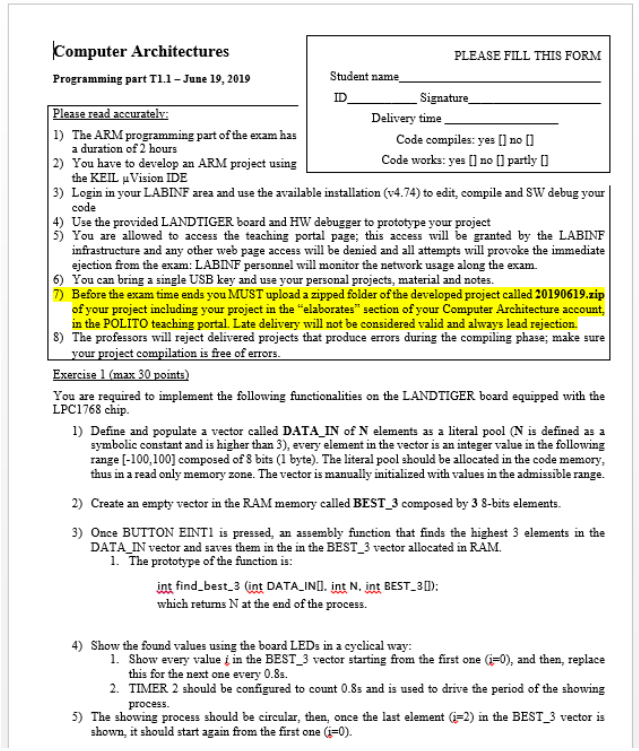
- Integer ALU: 1 clock cycle
- Data memory: 1 clock cycle
- FP multiplier unit: pipelined 10 stages
- FP arithmetic unit: pipelined 6 stages
- FP divider unit: not pipelined unit that requires 12 clock cycles
- branch delay slot: 1 clock cycle, and the branch delay slot is not enable
- forwarding is enabled
- it is possible to complete instruction EXE stage in an out-of-order fashion.

o and using the following code fragment, show the timing of the presented loop-based program and compute how many cycles does this program take to execute?

```
;*********************** MIPS64 ***********************
;     for (i = 0; i < 100; i++) {
;           v5[i] = v1[i]/v2[i];
;           v6[i] = (v1[i]-v2[i])+(v3[i]/v4[i]);
; }
```

		comments	Clock cycles
	.data		
v1:	.double "100 values"		
v2:	.double "100 values"		
v3:	.double "100 values"		
v4:	.double "100 values"		
v5:	.double "100 zeros"		
v6:	.double "100 zeros"		
	.text		
main:	daddui r1,r0,0	r1← pointer	
	daddui r2,r0,100	r2 <= 100	
loop:	l.d f1,v1(r1)	f1 <= v1[i]	
	l.d f2,v2(r1)	f2 <= v2[i]	
	l.d f3,v3(r1)	f3 <= v3[i]	
	l.d f4,v4(r1)	f4 <= v4[i]	
	div.d f5,f1,f2	f5 ← v1[i]/v2[i]	
	s.d f5,v5(r1)	v5[i] ← f5	
	sub.d f7,f1,f2	f7 ← v1[i]-v2[i]	
	div.d f6,f3,f4	f6 ← v3[i]/v4[i]	
	add.d f1,f7,f6	f1 ← f7 + f6	
	s.d f1,v6(r1)	v6[i] ← f1	
	daddui r1,r1,8	r1 ← r1 + 8	
	daddi r2,r2,-1	r2 ← r2 - 1	
	bnez r2,loop		
	halt		
		Total	

ARM System-on-Chip
(part 2 – 2h)



Computer Architectures

Programming part T1.1 – June 19, 2019

PLEASE FILL THIS FORM

Student name_____

ID_____ Signature_____

Delivery time _____

Code compiles: yes [] no []

Code works: yes [] no [] partly []

Please read accurately:

1) The ARM programming part of the exam has a duration of 2 hours
2) You have to develop an ARM project using the KEIL µVision IDE
3) Login in your LABINF area and use the available installation (v4.74) to edit, compile and SW debug your code
4) Use the provided LANDTIGER board and HW debugger to prototype your project
5) You are allowed to access the teaching portal page; this access will be granted by the LABINF infrastructure and any other web page access will be denied and all attempts will provoke the immediate ejection from the exam: LABINF personnel will monitor the network usage along the exam.
6) You can bring a single USB key and use your personal projects, material and notes.
7) Before the exam time ends you MUST upload a zipped folder of the developed project called **20190619.zip** of your project including your project in the "elaborates" section of your Computer Architecture account, in the POLITO teaching portal. Late delivery will not be considered valid and always lead rejection.
8) The professors will reject delivered projects that produce errors during the compiling phase; make sure your project compilation is free of errors.

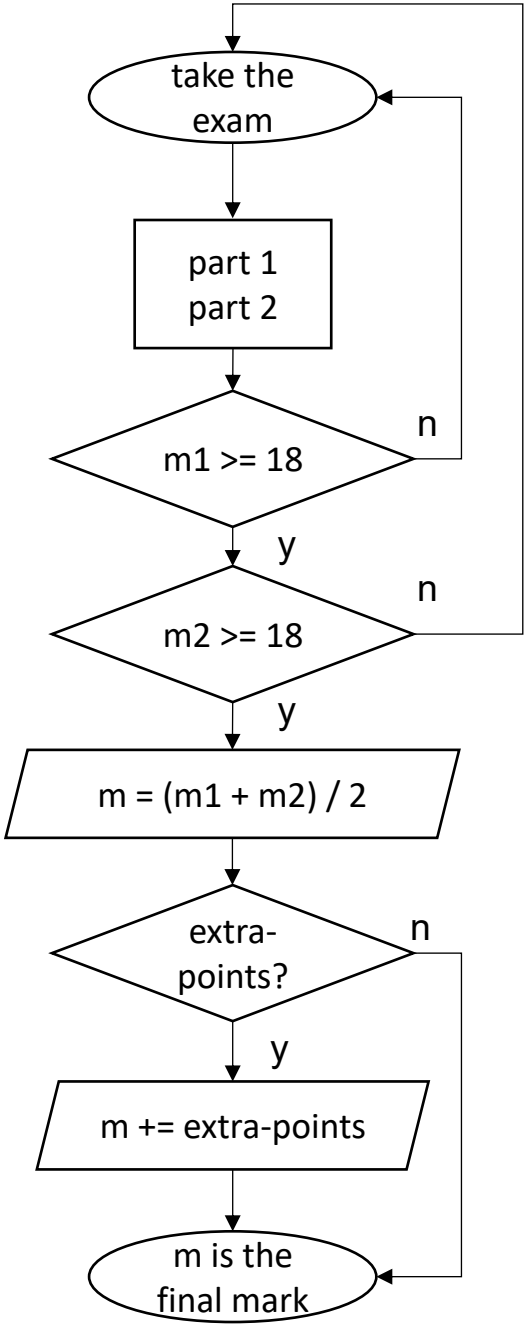
Exercise 1 (max 30 points)

You are required to implement the following functionalities on the LANDTIGER board equipped with the LPC1768 chip.

1) Define and populate a vector called **DATA_IN** of **N** elements as a literal pool (**N** is defined as a symbolic constant and is higher than 3), every element in the vector is an integer value in the following range [-100,100] composed of 8 bits (1 byte). The literal pool should be allocated in the code memory, thus in a read only memory zone. The vector is manually initialized with values in the admissible range.
2) Create an empty vector in the RAM memory called **BEST_3** composed by **3** 8-bits elements.
3) Once BUTTON EINT1 is pressed, an assembly function that finds the highest 3 elements in the DATA_IN vector and saves them in the in the BEST_3 vector allocated in RAM.
   1. The prototype of the function is:

      int find_best_3 (int DATA_IN[], int N, int BEST_3[]);

      which returns N at the end of the process.
4) Show the found values using the board LEDs in a cyclical way:
   1. Show every value *i* in the BEST_3 vector starting from the first one (i=0), and then, replace this for the next one every 0.8s.
   2. TIMER 2 should be configured to count 0.8s and is used to drive the period of the showing process.
5) The showing process should be circular, then, once the last element (i=2) in the BEST_3 vector is shown, it should start again from the first one (i=0).



Special projects that waive the exam

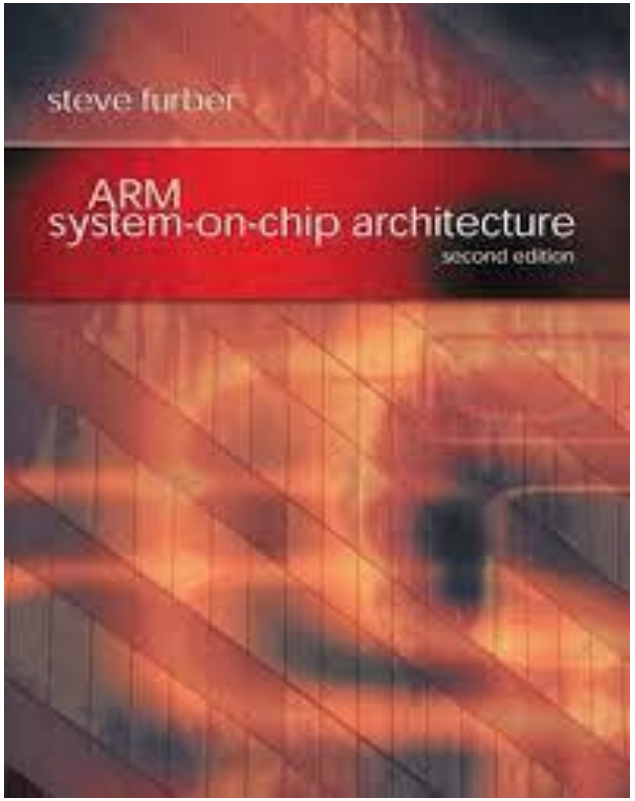
- By **January 27, 2026**, it will be possible to submit a candidature to be enrolled in a special project that waives the whole exam
- Eligibility criteria are the following:
  - The student must be attending the course for the first time
  - The student should have delivered all the lab solutions in time, including the final project
- Eligible students should obtain a positive outcome in an interview with the course professors
- Candidature is done by email, attaching a curriculum vitae.

Textbooks (I)

- J.L. Hennessy, D.A. Patterson
  Computer Architecture: a Quantitative Approach
  Morgan Kaufmann Publishers, Inc., V Edition, 2012

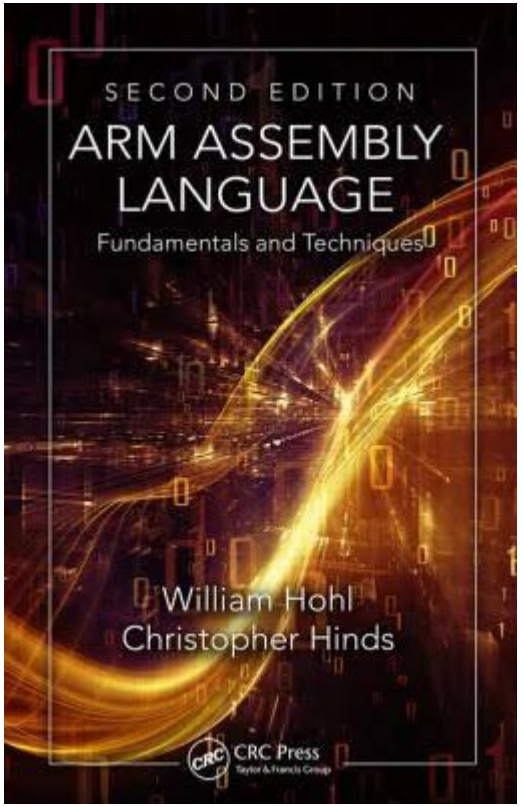
Textbooks (III)

- S. Furber
- *ARM – System-on-chip architecture*
- Addison-Wesley, 2000, II edition



Textbooks (IV)

- William Hohl, Christopher Hinds
- *ARM Assembly Language: Fundamentals and Techniques*
- CRC Press, 2014 II edition



Textbooks (V)

- Joseph Yiu
- *The definitive guide to the ARM Cortex-M3*
- Elsevier, II edition

Website

- Students may access to the course page through the institutional course site hosted by “il portale della didattica”:
- https://didattica.polito.it/
- where they can find:
  - Transparencies
  - Lab tracks
  - Announcements
  - General information
  - Recorded lectures
- Please take a look to cas.polito.it referring to the previous edition of the course (Computer Architectures, academic year 2018-19).

Communications

- Students are requested to check periodically the course web site and their official email address for possible communications by the teachers.
- It is kindly requested to the students to label the email subject using:
  - [ASE AAA-GRA] XXXX
  - [ASE G-Z] XXXX
  - [CAs] XXXXXX