Tecnológico
de Monterrey

Tecnológico de Monterrey

TE2015

Microcontroladores

Tecnológico de Monterrey

OUTLINE

- TE2015 Syllabus
- The history of the microcomputer
- Microchip fabrication process
- The Von Neumann architecture
- The Harvard architecture

Tecnológico de Monterrey

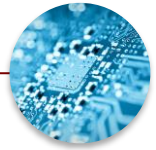
TE2015 Microcontroladores

Syllabus

ABOUT THE PROFESSOR

Researcher at Technical University of Denmark

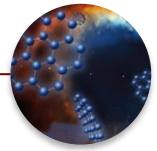
Adjunct professor at Tecnológico de Monterrey



Cofounder & CTO at AdSensors S.A.P.I. de C.V.

Teaching

- Microcontrollers, Digital Systems, Computer Architectures, Embedded Systems



Research interests

- Van der Waals Heterostructures & Electrokinetics
- UVM hardware Verification

Contact

- WhatsApp: +45 5033-3514
- E-mail: matias.vazquez@tec.mx

RESEARCH PROJECTS

Graphene-hBN Van der Waals Heterostructures (Technical University of Denmark)

- Graphene stacking and encapsulation
- Microfluidics & Electrokinetics
- Challenges related to TE2015
  - Pico-liter syringe pump
  - Vacuum control in desiccator
  - Precise UV radiation in flood exposure chamber

Hardware verification of NAND Flash Controller (SAGE Microelectronique, Tecnológico de Monterrey & Hangzhou Dianzi University)

- SystemVerilog testbench development under UVM framework
- Master students at Tec de Monterrey

Tecnológico de Monterrey

TO DOs BEFORE TUESDAY CLASS

Slack

- Install Slack app on your mobile or PC/Mac
- Join our group on Slack from the invitation e-mail received yesterday

EAGLE

- Create an account on Autodesk (use @tec.mx e-mail)
- Join our Autodesk group from the invitation e-mail received yesterday
- Download and install EAGLE

MPLAB X

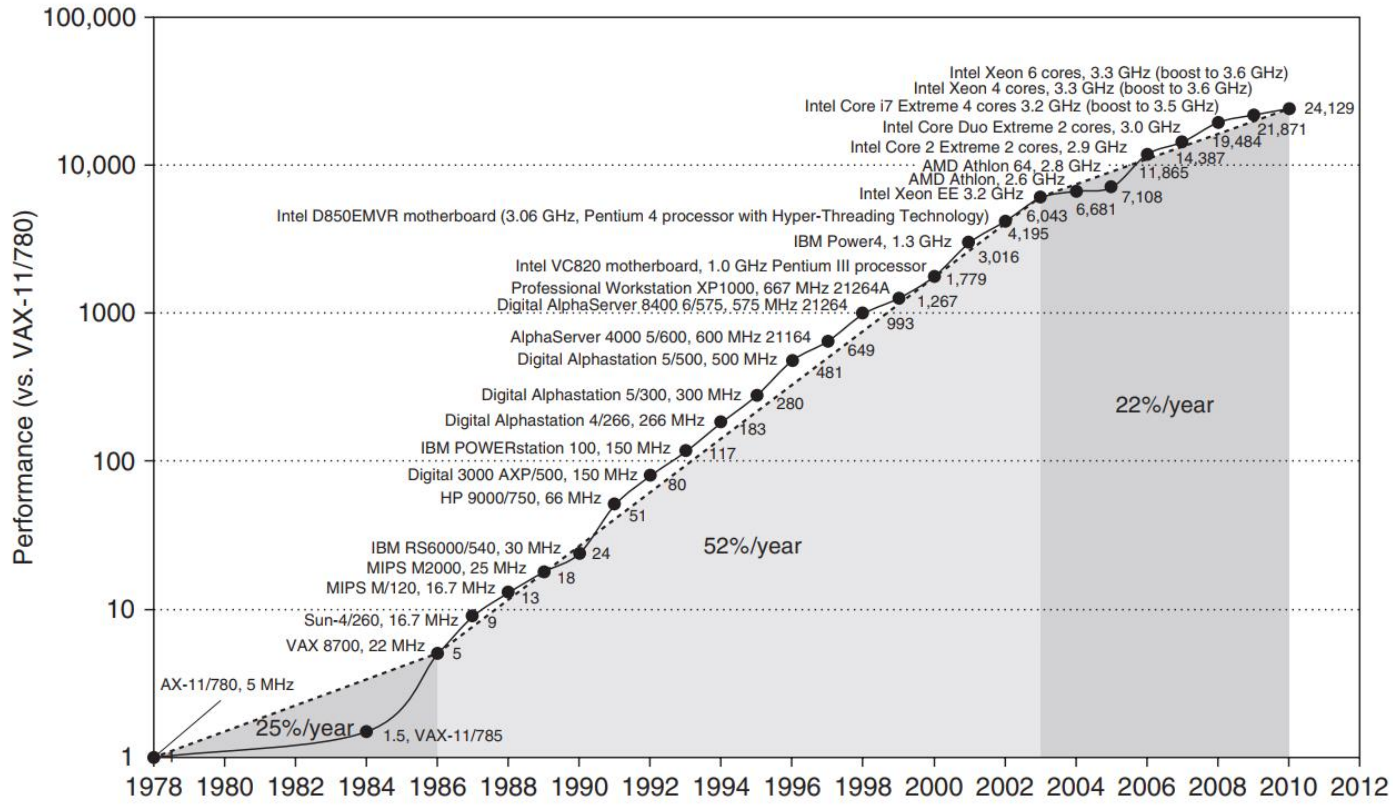
- Download MPLAB X from Microchip website and install it
- Download XC8 C compiler from Microchip website and install it

GitHub

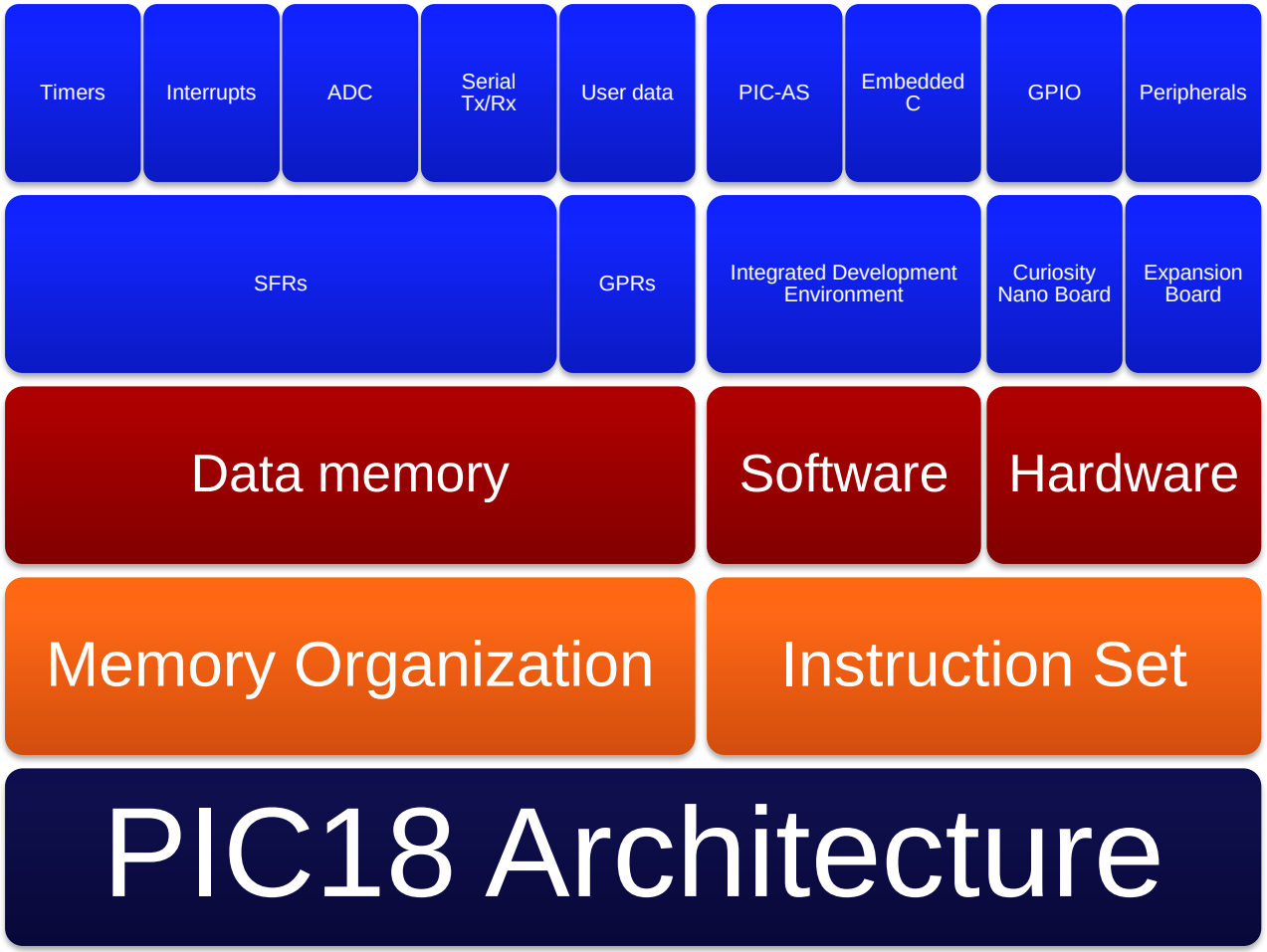
- Open an account on GitHub (use @tec.mx e-mail)
- Create a repository for the class
- Download and install GitKraken
- Make sure you can push your files from GitKraken to your repository

WHY A MICROCONTROLLERS CLASS?

- APPLICATION
- ALGORITHMS
- PROGRAMMING LANGUAGES
- OPERATING SYSTEMS/VIRTUAL MACHINES
- INSTRUCTION SET ARCHITECTURE
- MICROARCHITECTURE
- REGISTER-TRANSFER LEVEL
- LOGIC GATES
- ELECTRICAL CIRCUITS
- SEMICONDUCTOR DEVICES
- PHYSICS



CLASS SCOPE



1st Term

1. Introduction to microcontrollers
2. PIC18 microcontroller architecture
3. Memory organization in microcontrollers
4. Assembly language for PIC18

2nd Term

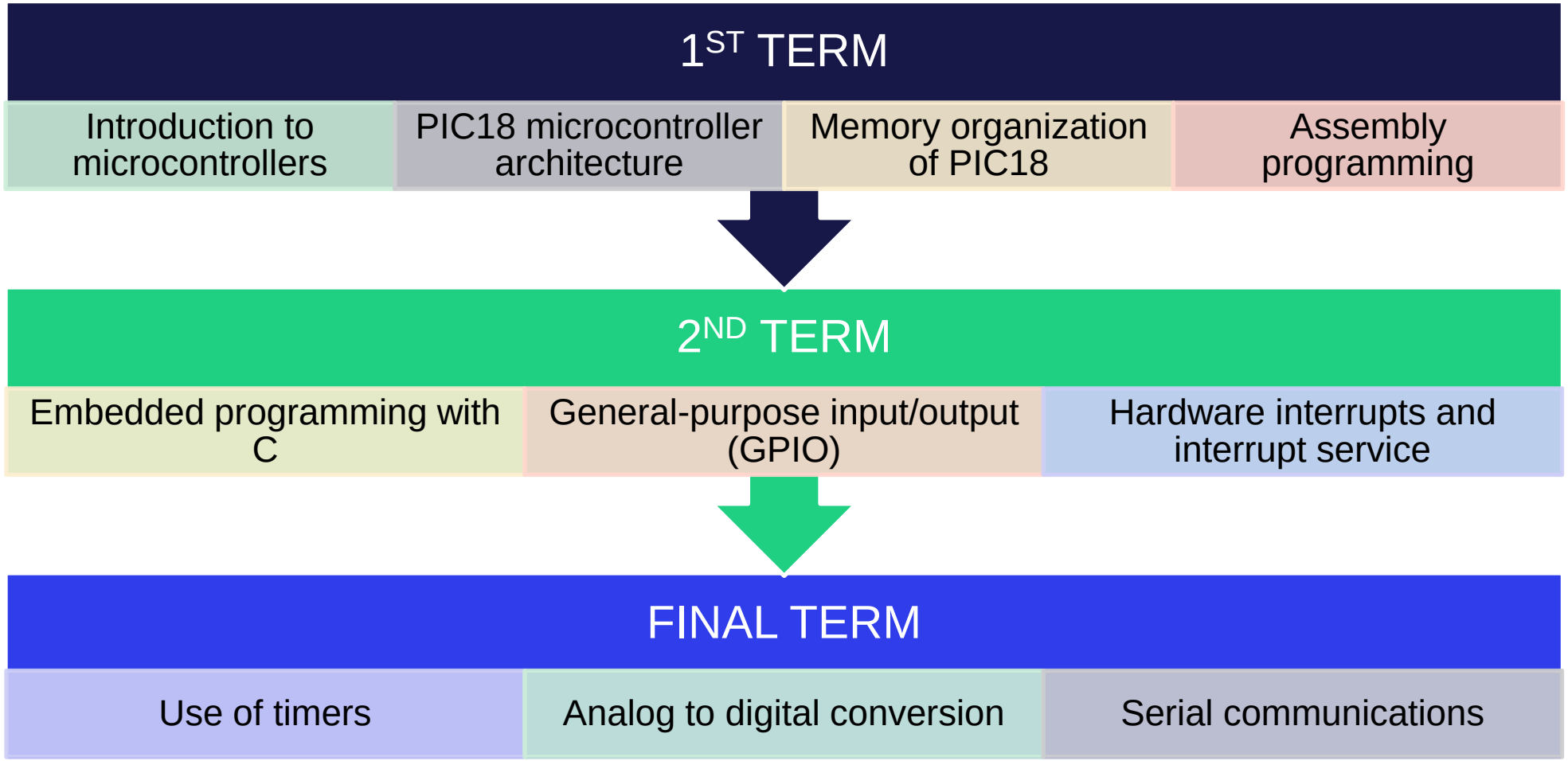
1. Embedded programming with C
2. General-purpose input/output (GPIO)
3. Hardware interrupts and interrupt service

Final Term

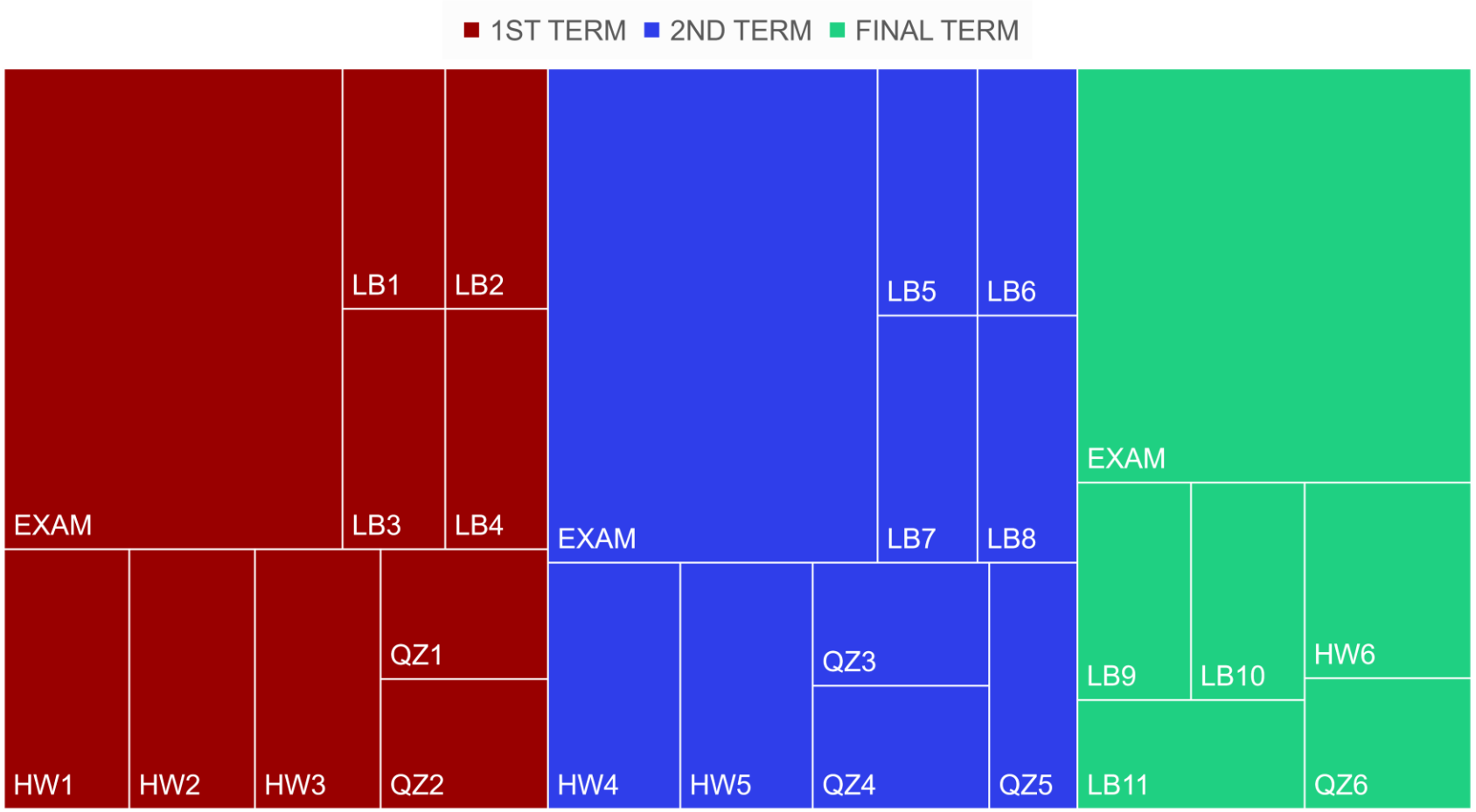
1. Use of timers
2. Analog to digital conversion
3. Serial communications

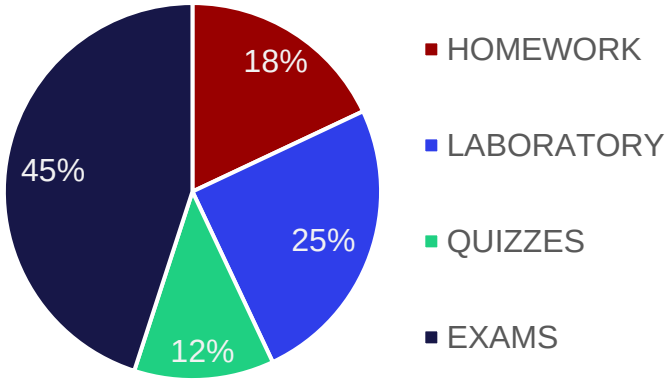
Tecnológico de Monterrey

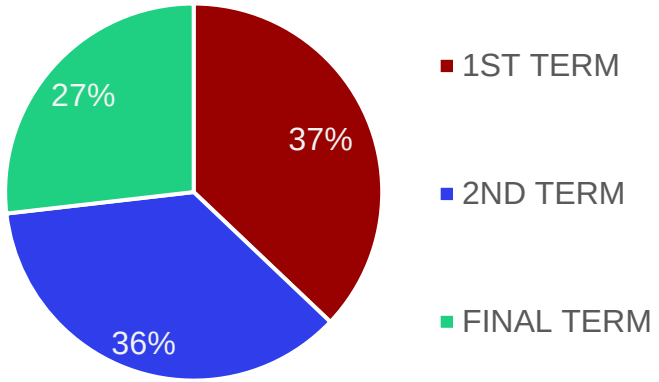
CLASS TOPICS



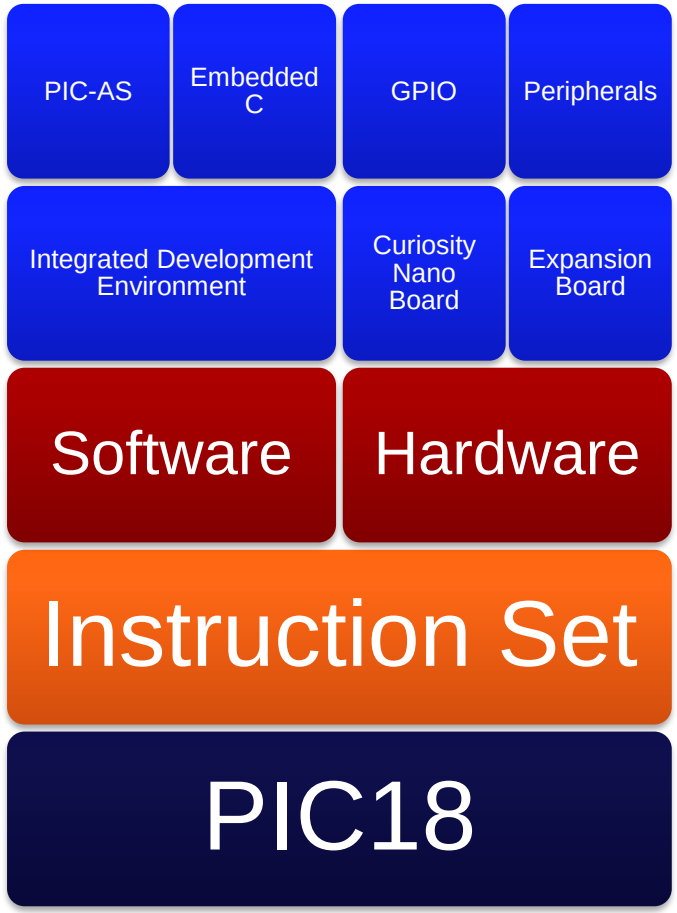
CLASS ACTIVITIES



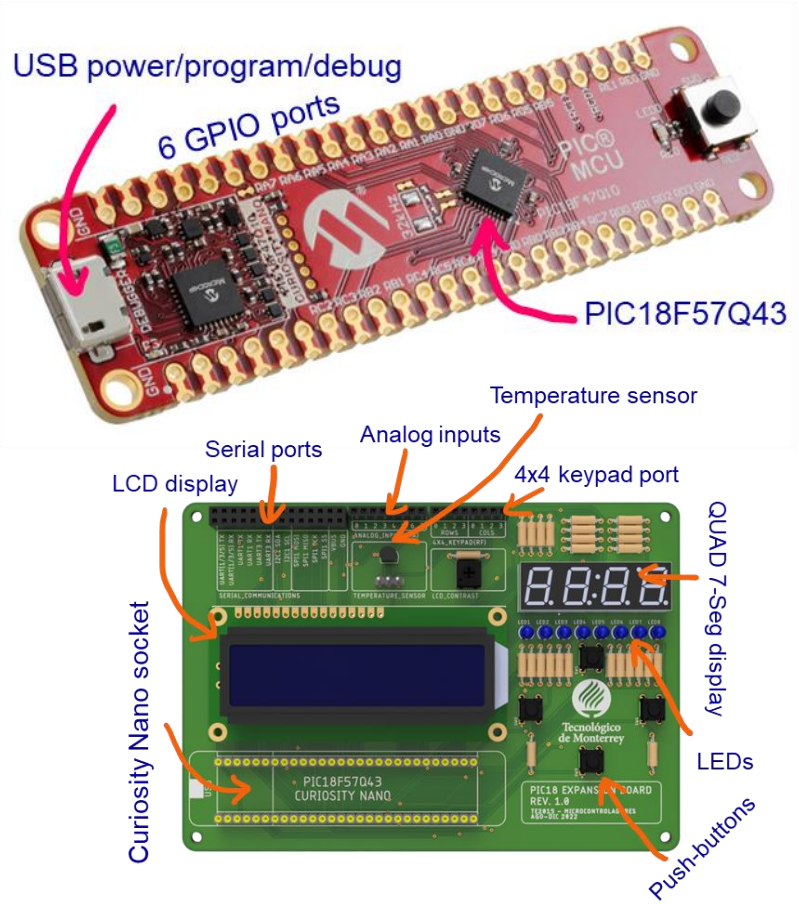




LABORATORY



WEEK	DATE	ACTIVITY	DESCRIPTION
1	08-Aug-22	LB1	Hardware design: schematic capture
2	15-Aug-22	LB2	Hardware design: physical layout
3	22-Aug-22	LB2	Hardware design: physical layout
4	29-Aug-22	LB3	Introduction to MPLAB and PIC18
5	05-Sep-22	LB4	Assembly programming of PIC18
6	12-Sep-22	LB4	Assembly programming of PIC18
7	19-Sep-22	LB5	High-level programming of PIC18 with C
8	26-Sep-22	LB5	High-level programming of PIC18 with C
9	03-Oct-22	LB6	GPIO: LCD display
10	10-Oct-22	LB7	GPIO: 4x4 keypad and 7-segment display
11	17-Oct-22	LB8	Hardware interrupts and ISR
12	24-Oct-22	LB8	Hardware interrupts and ISR
13	31-Oct-22	LB9	Timers, comparators and PWM
14	07-Nov-22	LB10	Analog inputs for microcontrollers
15	14-Nov-22	LB11	Asynchronous serial communications
16	21-Nov-22		HOLIDAY



LABORATORY

11 laboratories

25% of class grade

3 students per team

- Self-enrollment on Canvas

2 deliverables

- 50% Demostrative video (maximum length of 3 min.)
- 50% MPLAB project

CLASS RESOURCES

Integrated Develpment Environment

- XC8 Compiler
- MPLAB Xpress IDE

Bibliography

- PIC Microcontroller: An Introduction to Software & Hardware Interfacing - Huang
- Interfacing PIC Microcontrollers to Peripheral Devices – Bohdan Borowik
- Programming 8-bit Microcontrollers in C – Martin P. Bates
- C Programming for the PIC Microcontroller – Hubert Henry Ward

PIC18 & MPLAB Documentation

- MPLAB XC8 PIC Assembler
- MPLAB C18 C Compiler
- PIC18F57Q43 Datasheet
- CURIOSITY NANO PIC18F57Q43 User Guide

GitHub Repository

- Laboratories
- Datasheets
- Examples
- Simulations

Slack

- Group Communications
- Questions & Queries
- File sharing
- Team channels

Tecnológico de Monterrey

TE2015 Microcontroladores

The history of the microcomputer

Invención del Transistor

- Inventado en Bells Lab
- Hecho de Germanio
- Junta bipolar tipo NPN

1948

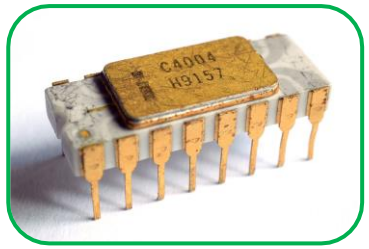
1966

- **16 Bits**
- Diseñada en el MIT
- Usada para guía, navegación y control del modulo lunar del Apollo 11.

Apollo guidance computer

Intel 4004

- **4 Bits, proceso MOS**
- Primer µprocesador commercial
- Diseñado para calculadoras



1971

THE C PROGRAMMING LANGUAGE

1975

- **8 Bits**
- Microprocesador de bajo costo
- Inició la revolución de la computación en casa

MOS 6502

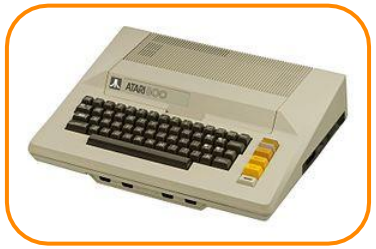
Apple 1

- Incluía un MOS 6502
- Fabricaron 200 modelos
- Diseñada para uso recreativo

1976

1979

- Incluía un MOS 6502
- Diseñadas para juegos y computadora personal
- Integraba un teclado completo



Atari 400 y 800

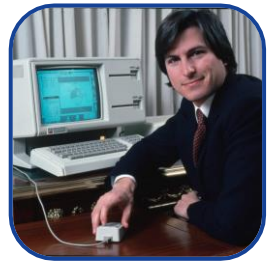
IBM PC (5150)

- Procesador INTEL 8088 de 16 Bits
- SO Microsoft MS-DOS
- Introdujo el concepto de Computadora Personal

1981

1983

- Motorola 68000 de 16/32 Bits
- SO Lisa OS
- Primera PC con interfaz gráfica.
- Fracaso comercial



Apple Lisa

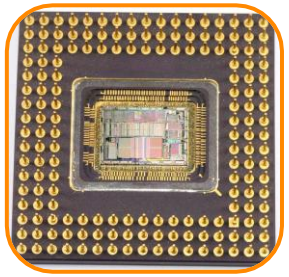
Compaq Deskpro 386

- Procesador Intel 80386 de 32 Bits
- BUS de datos de 16 Bits
- SO Windows/386.

1986

1989

- Procesador de 32 Bits
- BUS de datos mejorado
- Doblada el desempeño respecto a la 386



Intel 80486

Apple PowerBook

- Motorola 68000 de 16/32 Bits
- SO System 7.0.1
- Considerada como una de las computadoras más importantes de todos los tiempo

1991

1995

- Procesador R3000A de 32 Bits
- Creada en colaboración con Nintendo para desarrollar una versión de CD-ROM del Súper Nintendo

Sony Playstation

PowerMac G5

- Procesador PowerPC 970 de 64 Bits
- SO MacOS X.
- Primera PC de 64 bits

2003

Apple iPhone

- Procesador de 32 Bits* (Samsung ARM11)
- Combinaba web browsing, reproducción de música, y teléfono celular

2007

Microsoft Xbox One

- AMD Jaguar CPU con dos módulos Quad-Core, sumando 8 núcleos de 64 bits
- Soportaba 1080p y 720p

2013

Intel Core i9

- Procesador de 10 núcleos de 64 Bits
- Incorpora 7 mil millones de transistores en un volumen de 37.5 mm × 37.5 mm × 4.4 mm

2018

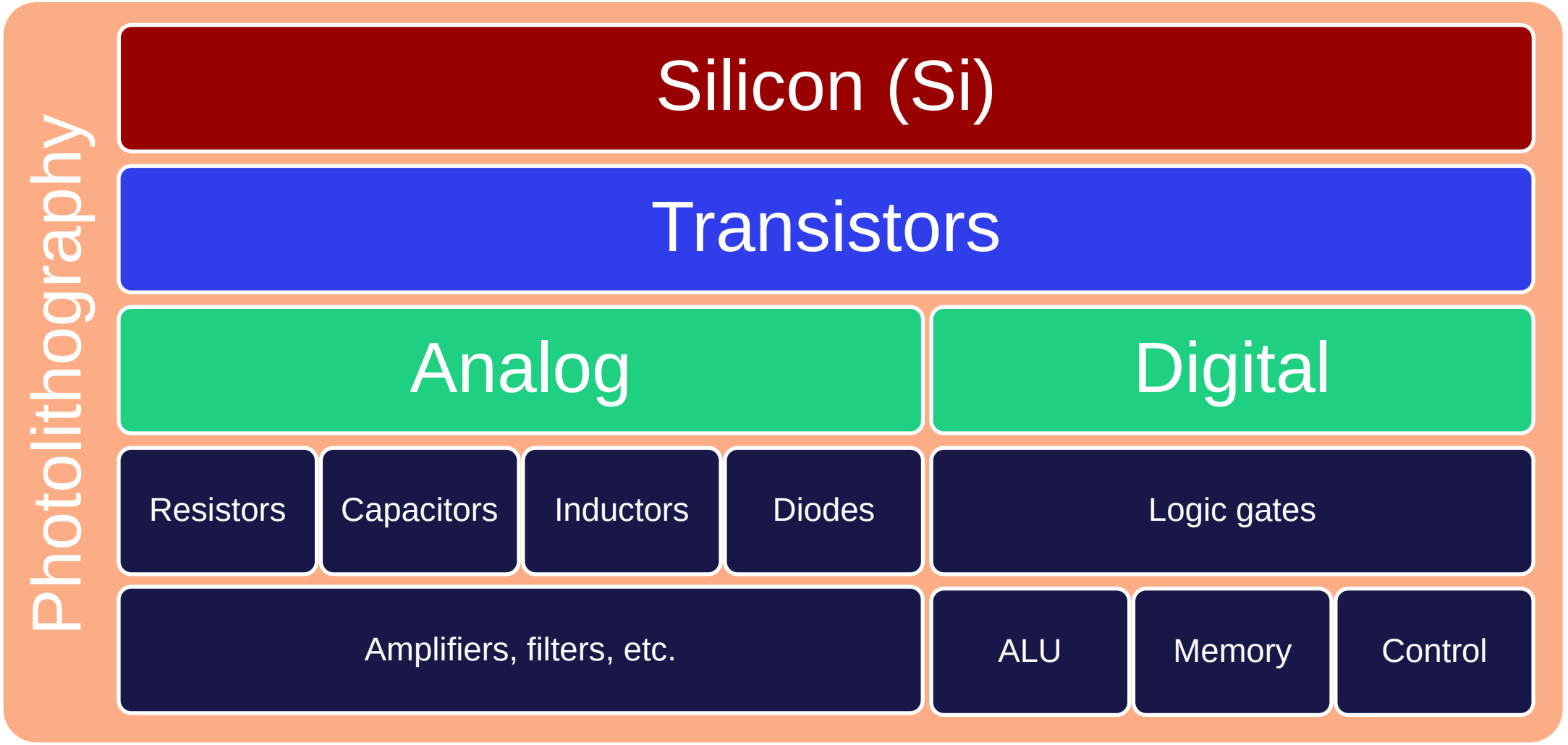
Tecnológico de Monterrey

TE2015 Microcontroladores

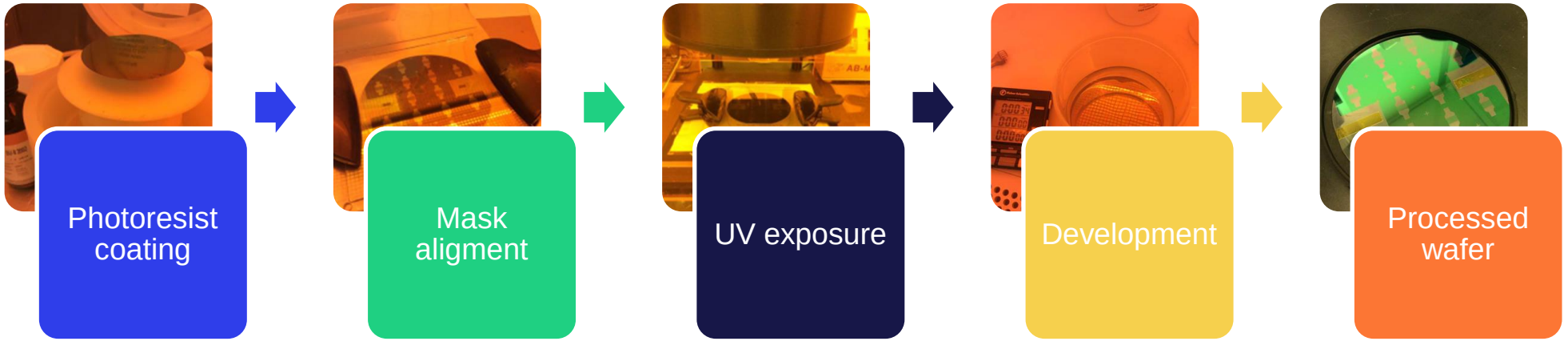
Microchip fabrication process

Tecnológico de Monterrey

SEMICONDUCTORS FABRICATION PROCESS



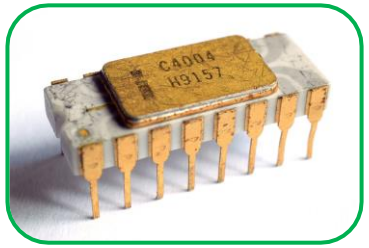
SEMICONDUCTORS FABRICATION PROCESS

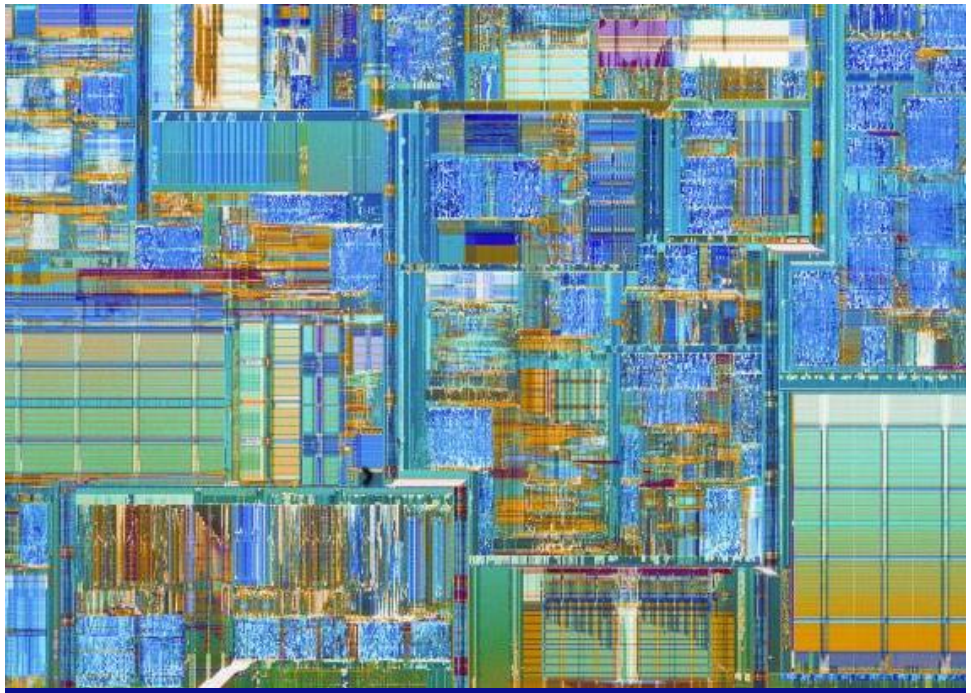



intel

Tecnológico de Monterrey

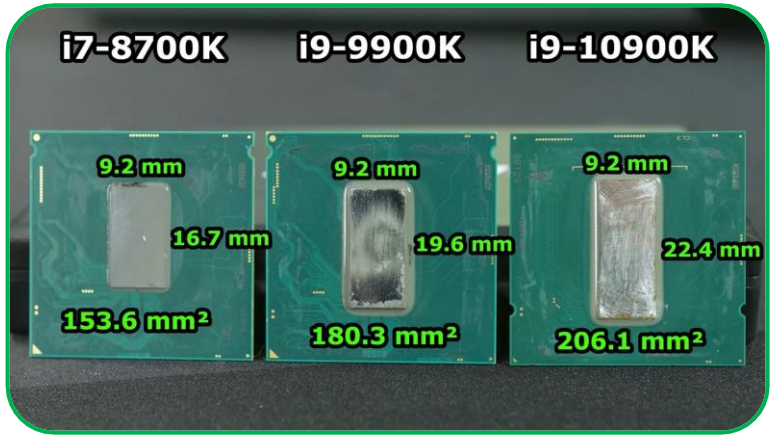
INTEL 4004 (1971)

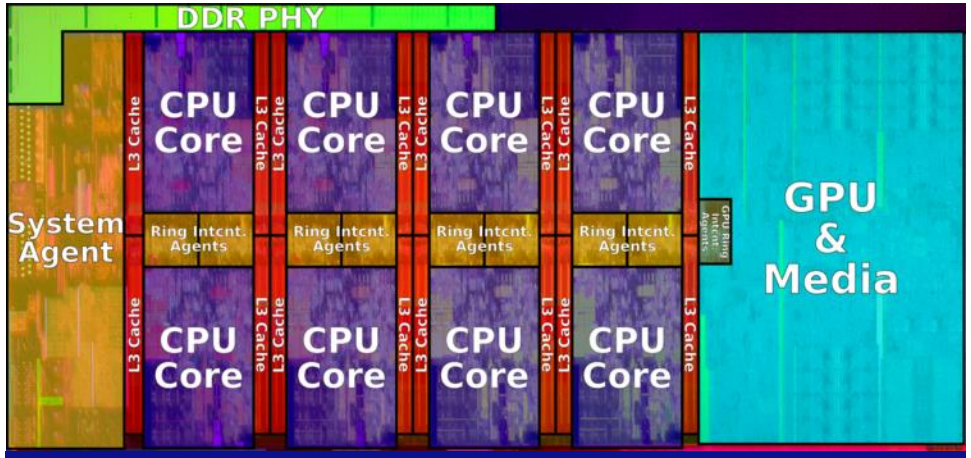




2300 transistors in 4 mm x 3 mm
192 transistors/mm^2

INTEL CORE i9-9900K (2018)





~7 billion transistors in 9.2 mm x 19.6 mm
39 million transistors/mm^2

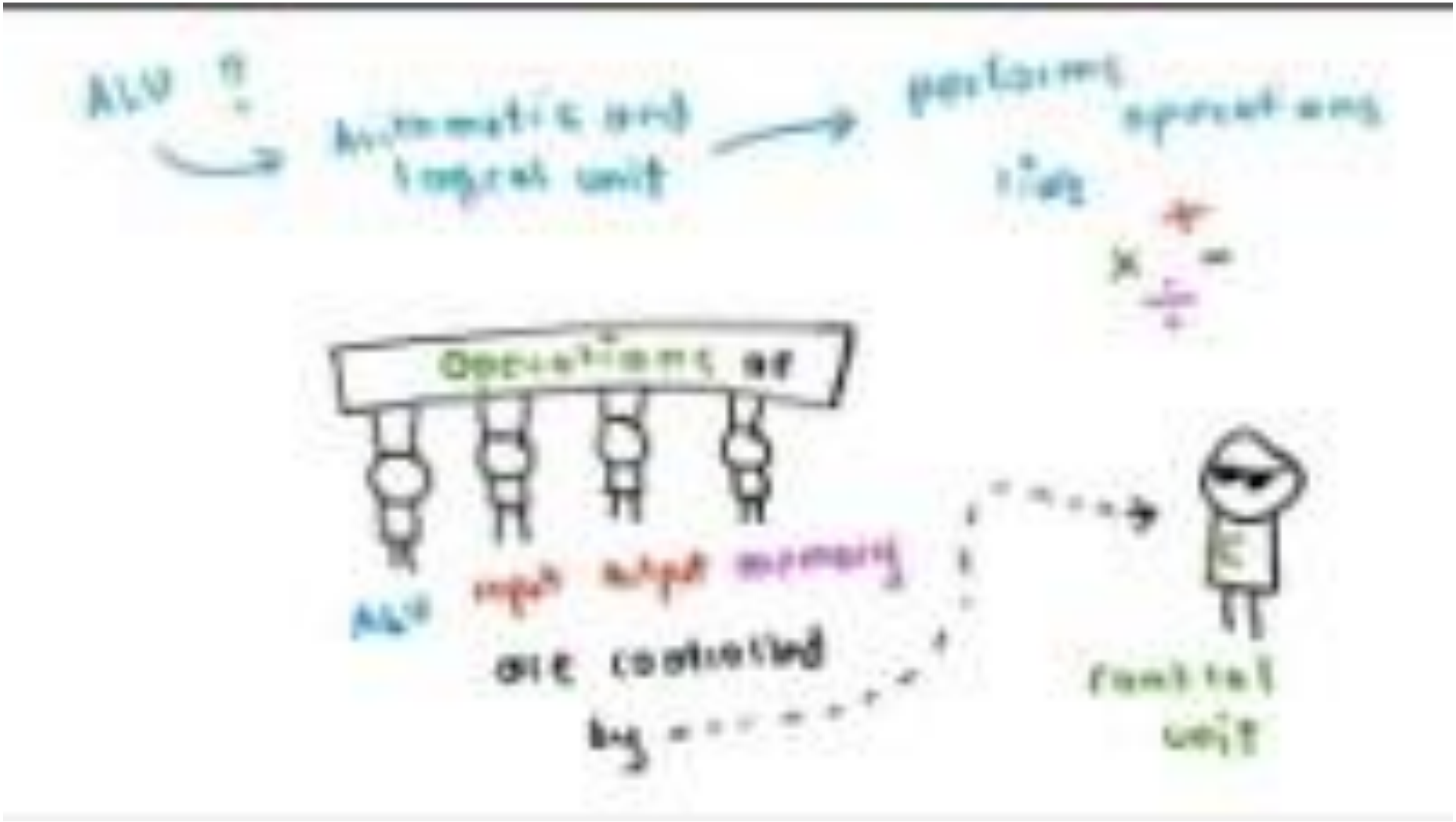
Tecnológico de Monterrey

TE2015 Microcontroladores

Von Neumann and Harvard Architectures

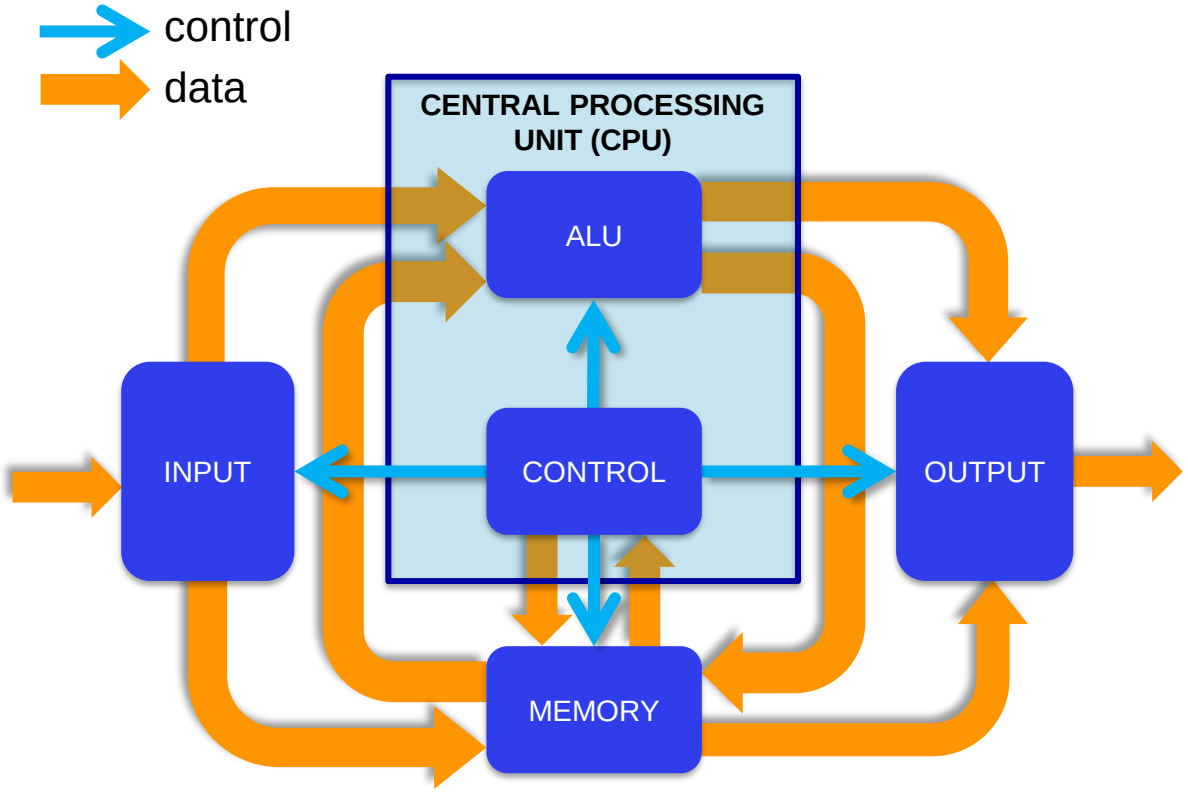
Tecnológico de Monterrey

https://bit.ly/TE2015MIRO



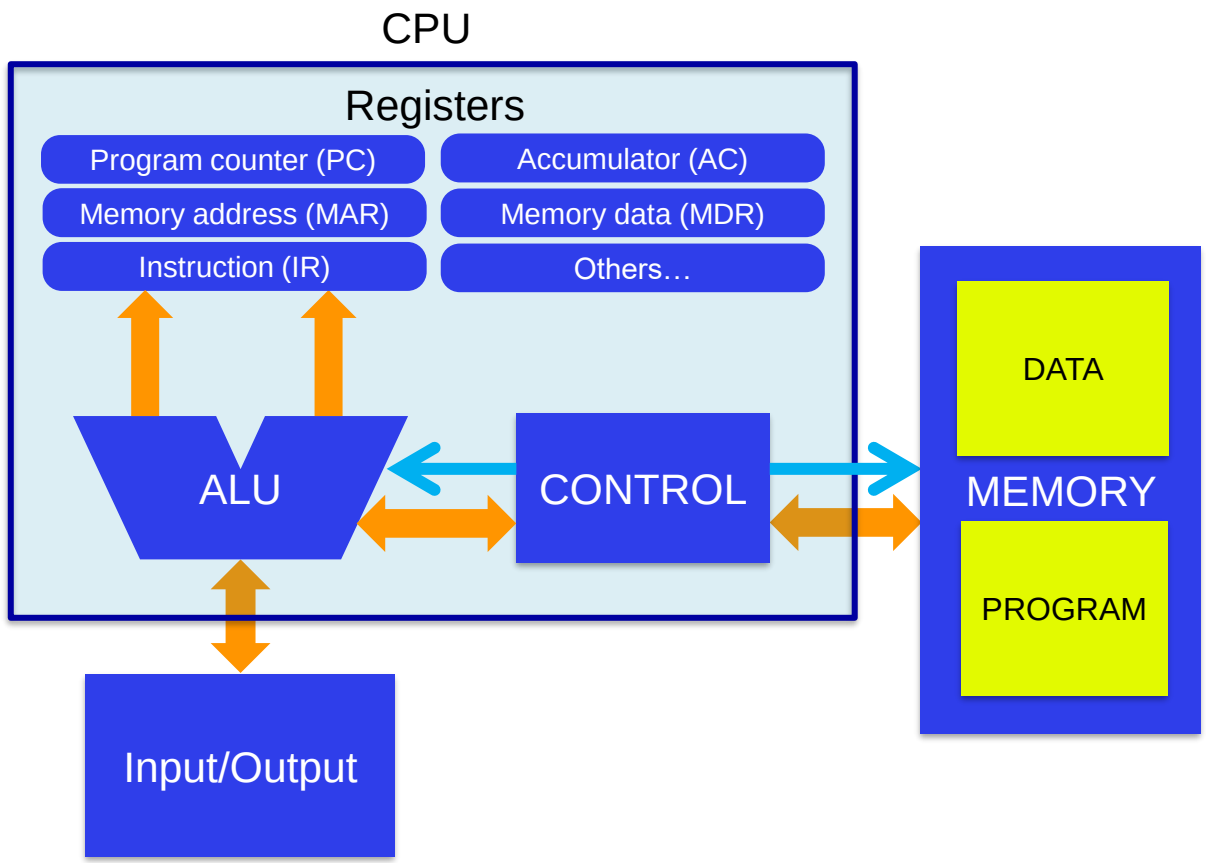
https://youtu.be/5BpgAHBZgec

VON NEUMANN ARCHITECTURE



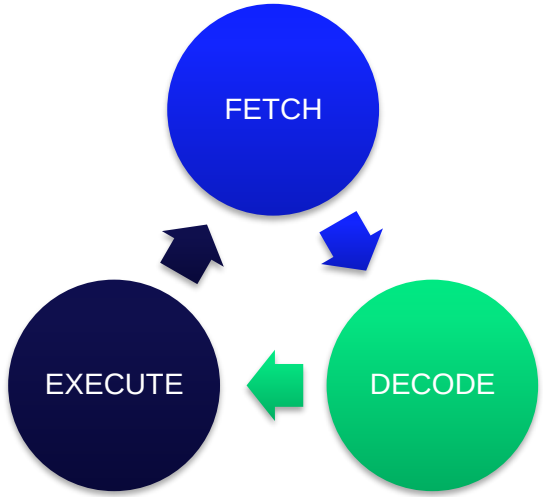
- The CPU includes the ALU and Control units, plus some Registers
- Memory address bus is implicit in data bus between Control and Memory
- Registers are small memories built into the CPU for fast access to data (low latency)
- The main registers in this architecture are
  - Program counter (PC)
  - Accumulator (AC)
  - Memory address (MAR)
  - Memory data (MDR)
  - Instruction (IR)
- Main memory holds instructions and data
- The Input/Output unit is a signal conditioning stage for incoming/outcoming data

VON NEUMANN ARCHITECTURE

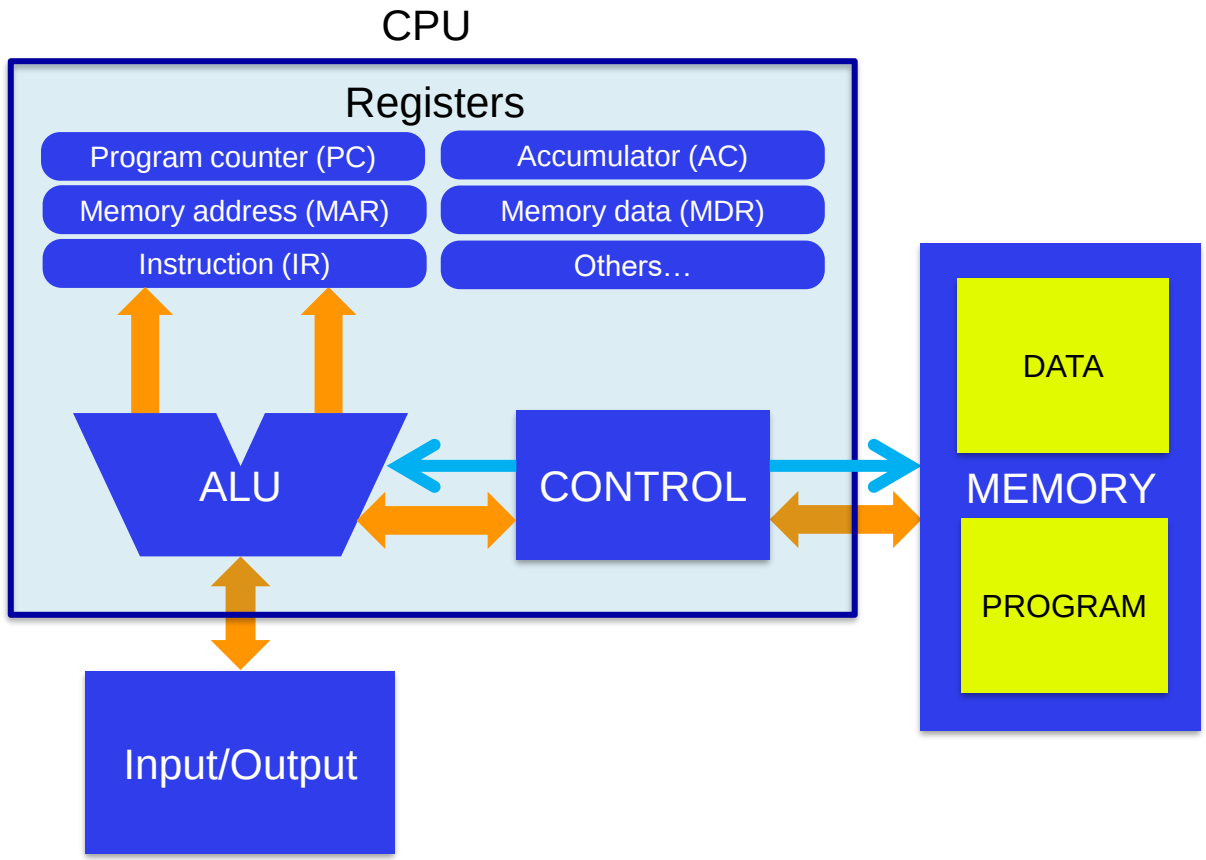


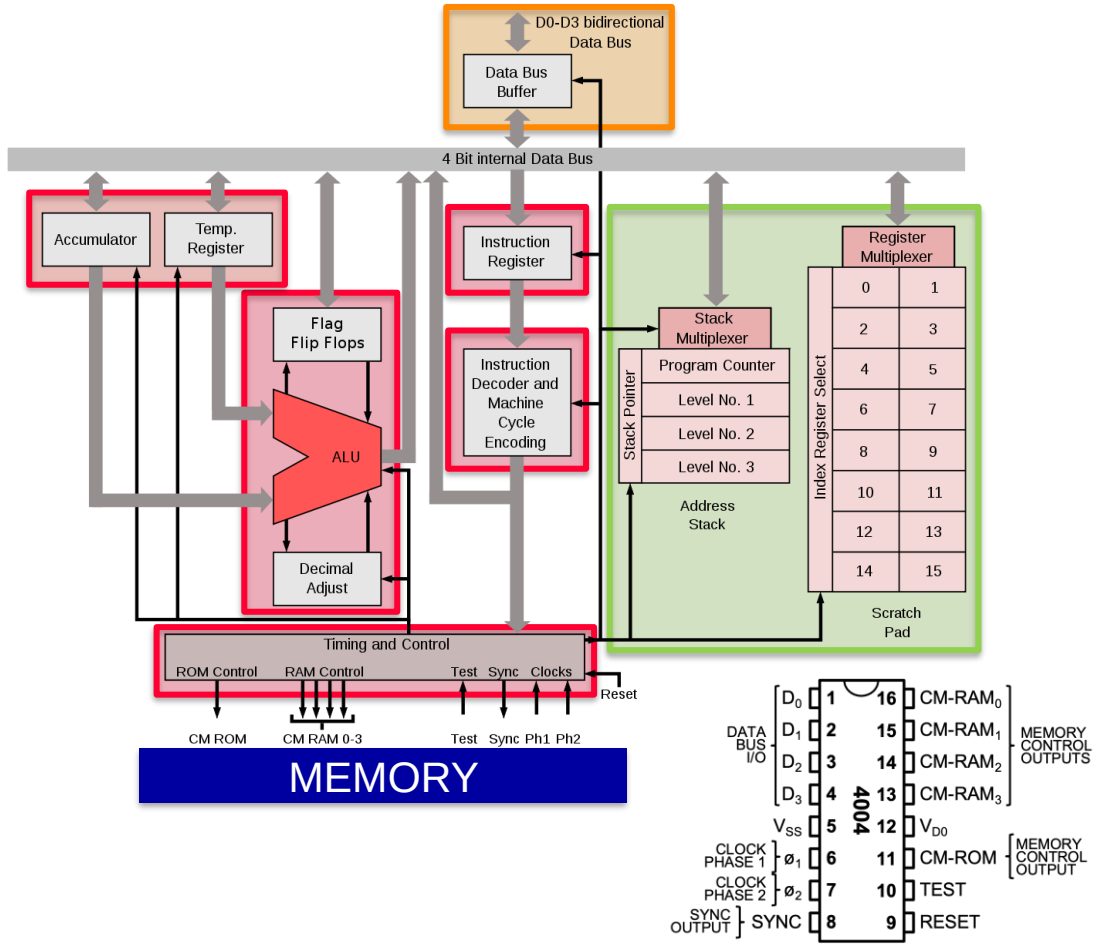
INSTRUCTION CYCLE

1. The Control unit **fetches** the next program instruction from main memory using the PC to determine where the instruction is located
2. The instruction is **decoded** so the ALU can understand it
   - Data operands required to execute the instruction are fetched from main memory and placed in registers in the CPU
3. The ALU **executes** the instruction and stores the result in registers or main memory

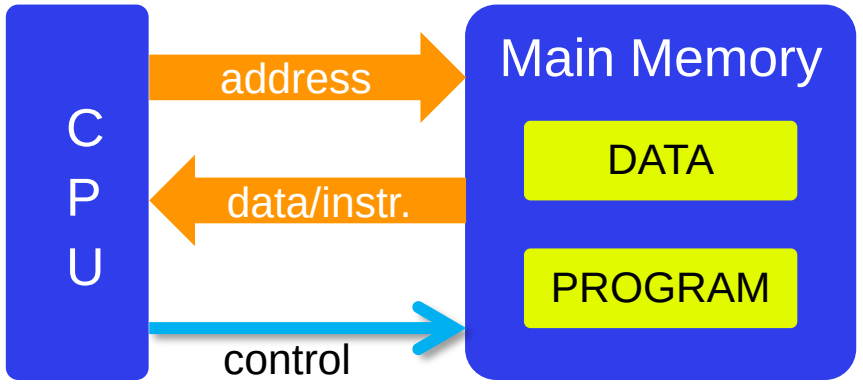


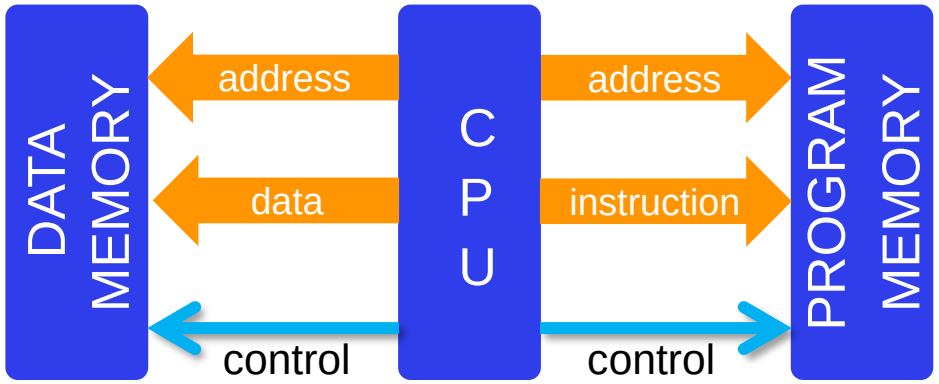
VON NEUMANN EXAMPLE: INTEL 4004





VON NEUMANN VS HARVARD





VON NEUMANN

- Data and Program in the same physical memory
- Single BUS that **sequentially** reads data and instructions
  1. One instrution cycle to read an instruction
  2. One instruction cycle to read data
- **Slower but simpler than Harvard**

HARVARD

- Data and program in separate physical memory
- **Fetch** can be carried out while other instruction is in **Execution**
  - This is called **Pipelining**
- **Faster but more complex than Von Neumann**