Computer Engineering

ECC301

Microprocessors

CH1

Syllabus

Credits: 4

Prereq.:

ECC203 Computer Architecture and Organization

Objectives of the Course:

Teaching the microprocessor as a programmable digital system element. To illustrate some basic concepts of microprocessors through the use of assembly language programming. To give the principles of hardware design; To provide an understanding of a microprocessor based system as a combination of hardware and software subsystems and their interactions.

Syllabus

Course Description:

Introduction to microprocessors. Architecture of microprocessors and instruction sets. Interrupts. Memories. Parallel and serial input/output programming. Microprocessor based system design. Microprocessors applications.

Grading:

- Lab 20%
- Project 30%
- Final Exam 50%

Syllabus

Textbooks

- Alexandru Radovici and Ioana Culic, *Getting Started with Secure Embedded Systems: Developing IoT Systems for micro:bit and Raspberry Pi Pico Using Rust and Tock*, Apress, 2022.
- Dogan Ibrahim, *Raspberry Pi Pico Essentials*, Elektor, 2021.
- Gareth Halfacree and Ben Everard, Get started with MicroPython on Raspberry Pi Pico,2/e, Raspberry Pi, 2021. (download)
- KCS Murti, *Design Principles for Embedded Systems*, Springer, 2022.
- Stephen Smith, *RP2040 Assembly Language Programming ARM Cortex-M0+ on the Raspberry Pi Pico*, Apress, 2022.
- Dogan Ibrahim and Kaan Uyar, *The 8080 and 8085 Microprocessors and Peripherals*, Bilesim Yayincilik, 2006
- Dogan Ibrahim and Kaan Uyar, *8085 Microprocessor Experiments*, Bilesim Yayincilik, 2006
- Barry B. Brey, *The Intel microprocessors*, 8e, Pearson, 2014

Syllabus

Online resources

- https://www.arm.com/
- https://www.raspberrypi.com
- https://micropython.org
- others

Microprocessor any of a type of miniature electronic device that contains the arithmetic, logic, and control circuitry necessary to perform the functions of a digital computer’s central processing unit. In effect, this kind of integrated circuit can interpret and execute program instructions as well as handle arithmetic operations.

https://www.britannica.com/technology/microprocessor

History

Transistor was invented in 1948 (23 December 1947 in Bell lab) by three American physicists, ***John Bardeen***, ***Walter H. Brattain***, and ***William B. Shockley.*** https://www.britannica.com/technology/transistor

Integrated circuit (IC) was invented in 1958 by electrical engineer ***Jack Kilby*** of Texas Instruments / physicist ***Robert Noyce*** of Fairchild Semiconductor Corporation. https://www.britannica.com/technology/integrated-circuit

INTEL (INTegrated ELectronics) was founded in July 1968 by ***Robert Noyce*** and ***Gordon Moore***. ***First microprocessor*** was invented by Intel in 1971. They named it **Intel 4004** because it was a 4-bit microprocessor. https://www.britannica.com/topic/Intel

Computer Architecture

RISC

A Reduced Instruction Set Computer is a type of architecture that utilizes a small, highly-optimized set of instructions rather than the highly-specialized set of instructions typically found in other architectures.

ARM, PowerPC, Sparc, RISC-V...

CISC

A Complex Instruction Set Computer is a type of architecture in which single instructions can execute several low-level operations or are capable of multi-step operations or addressing modes within single instructions.

×86 CPUs, Zilog, Motorola…

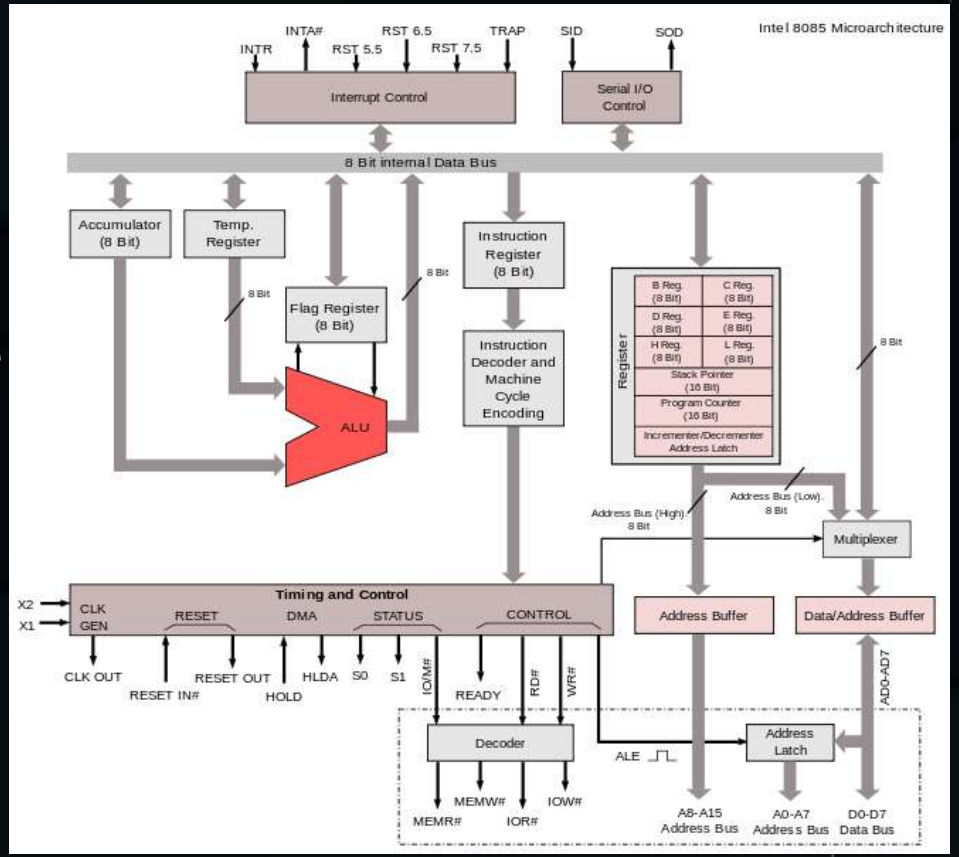
Intel

Name	Year of Invention	Clock speed	Size of microprocessor	Number of transistors	Inst. per sec
INTEL 4004/4040	1971	740 KHz	4 bit	2300	60000
8008	1972	500 KHz	8 bit	3500	50000
8080	1974	2 MHz	8 bit	6000	500000
8085	1976	3 MHz	8 bit (16 bit address bus)	6500	769230
8086	1978	4.77 MHz, 8 MHz, 10 MHz	16 bit (multiply and divide instruction, 16 bit data bus and 20 bit address bus)	29000	2.5 Million
8088	1979		16 bit (cheaper version of 8086 and 8 bit external bus)		2.5 Million
80186/80188	1982	6 MHz	16 bit (80188 cheaper version of 80186, and additional components like interrupt controller, clock generator, local bus controller, counters)		
80286	1982	8 MHz	16 bit (data bus 16bit and address bus 24 bit)	134000	4 Million
80386	1986	16 MHz-33 MHz	32 bit	275000	
...					

Intel

The 8085 is a conventional ***von Neumann design*** based on the Intel 8080.

The "5" in the part number highlighted the fact that the 8085 uses a single +5 V power supply by using depletion-mode transistors (Intel's enhanced nMOS process called HMOS II), rather than requiring the +5 V, −5 V and +12 V supplies needed by the 8080.



ARM

With RISC, a central processing unit (CPU) implements the processor design principle of simplified instructions that can do less but can execute more rapidly. The result is improved performance. A key RISC feature is that it allows developers to increase the register set and increase internal parallelism by increasing the number of parallel threads executed by the CPU and increasing the speed of the CPU's executing instructions.

ARM, or “Advanced RISC Machine” is a specific family of instruction set architecture that’s based on reduced instruction set architecture developed by Arm Ltd. Processors based on this architecture are common in smartphones, tablets, laptops, gaming consoles and desktops, as well as a growing number of other intelligent devices.

Arm Instruction Set Architecture

Cortex A

The Arm Application-profile (A-profile) architecture targets high-performance markets, such as PC, mobile, gaming, and enterprise which is required for full feature OS such as Linux, Android, Windows and mobile OSs.

The latest versions of the A-profile architecture are *Armv9-A* and *Armv8-A*.

See the Arm Cortex-A Processor Comparison Table

https://developer.arm.com/documentation/102826/latest/

Arm Instruction Set Architecture

Cortex R

The Arm Real-time profile (R-profile) provides high-performing processors for safety-critical environments. Although these processors cannot run full versions of Linux or Windows, there are plenty of *Real Time Operating Systems (RTOS)* that can be used with these processors.

The *Armv8-R* architecture is the latest generation Arm architecture targeted at the Real-time profile. It introduces virtualization at the highest security level while retaining the Protected Memory System Architecture (PMSA) based on a Memory Protection Unit (MPU). It supports the *A32* and *T32 instruction sets*.

Arm Instruction Set Architecture

Cortex M

The Arm Microcontroller profile (M-profile) provides low-latency, highly deterministic operation for deeply embedded systems.

Armv8.1-M architecture includes the M-Profile Vector Extension (MVE) that provides major uplift in levels of machine learning and signal processing performance. It implements the simplified programmer's model of Cortex-M processors, to bring advanced compute capabilities to millions of developers. In Arm Cortex-M processors, MVE is named *Arm Helium* technology. The architecture also enhances system-wide security with *Arm TrustZone*.

Arm Instruction Set Architecture

The Arm architecture supports three instruction sets: A64, A32 and T32.

- The A64 and A32 instruction sets have fixed instruction lengths of 32-bits.
- The T32 instruction set was introduced as a supplementary set of 16-bit instructions that supported improved code density for user code. Over time, T32 evolved into a 16-bit and 32-bit mixed-length instruction set. As a result, the compiler can balance performance and code size trade-off in a single instruction set.

Arm Instruction Set Architecture

A64 instruction set

The A64 instruction set is supported by the *Armv8-A architecture*. Key features of A64 include:

- Clean decode table based on 5-bit register specifiers.
- Instruction semantics broadly similar to A32 and T32.
- 31 general-purpose 64-bit registers accessible at all times.
- No modal banking of general purpose registers for improved performance and energy.
- Program counter and stack pointer are not general purpose registers.
- Dedicated zero register available for most instructions.

Arm Instruction Set Architecture

A32 Instruction Set

A32 instructions, known as Arm instructions in pre-Armv8 architectures, are 32 bits wide, and are aligned on 4-byte boundaries. A32 instructions are supported by both *A-profile* and *R-profile* architectures.

A32 was traditionally used in applications requiring the highest performance, or for handling hardware exceptions such as interrupts and processor start-up. Much of its functionality was subsumed into *T32* with the introduction of Thumb-2 technology.

Arm Instruction Set Architecture

T32 Instruction Set

The T32 instruction set, known as Thumb in pre-Armv8 architectures, is a mixed 32- and 16-bit length instruction set that offers the designer excellent code density for minimal system memory size and cost.

T32 provides enhanced levels of performance, energy efficiency, and code density for a wide range of embedded applications. Designers can use both T32 and A32 instructions sets and therefore have the flexibility to emphasize performance or code size on a subroutine level as their applications require. T32 is supported across all architecture profiles, and is the only instruction set supported by the *M-Profile* architecture.

The Cortex-M processor family

Cortex-M0

A very small processor (starting from 12K gates) for low cost, ultra low power microcontrollers and deeply embedded applications

Cortex-M0+

The most energy-efficient processor for small embedded system. Similar size and programmer's model to the Cortex-M0 processor, but with additional features like single cycle I/O interface and vector table relocations

The Cortex-M processor family

Cortex-M1

A small processor design optimized for FPGA designs and provides Tightly Coupled Memory (TCM) implementation using memory blocks on the FPGAs. Same instruction set as the Cortex-M0

Cortex-M3

A small but powerful embedded processor for low-power microcontrollers that has a rich instruction set to enable it to handle complex tasks quicker. It has a hardware divider and Multiply-Accumulate (MAC) instructions. In addition, it also has comprehensive debug and trace features to enable software developers to develop their applications quicker

The Cortex-M processor family

Cortex-M4

It provides all the features on the Cortex-M3, with additional instructions target at Digital Signal Processing (DSP) tasks, such as Single Instruction Multiple Data (SIMD) and faster single cycle MAC operations. In addition, it also have an optional single precision floating point unit that support IEEE 754 floating point standard.

Cortex-M7

High-performance processor for high-end microcontrollers and processing intensive applications. It has all the ISA features available in Cortex-M4, with additional support for double-precision floating point, as well as additional memory features like cache and Tightly Coupled Memory (TCM).

The Cortex-M processor family

Cortex-M23

A small processor for ultra-low power and low cost designs, similar to the Cortex-M0+ processor, but with various enhancements in instruction set and system-level features. It also supports the TrustZone security extension.

Cortex-M33

A mainstream processor design, similar to previous Cortex-M3 and Cortex-M4 processors, but with much better flexibility in system design, better energy efficiency and higher performance. It also supports the TrustZone security extension.

The Cortex-M processor family

Cortex-M35P

Cortex-M35P is the first Armv8-M processor with tamper-resistance built in to bring physical security within reach of any IoT product. Cortex-M35P has the option of integrated digital signal processing (DSP), SIMD, and multiply-accumulate (MAC) instructions that simplify overall system design, software development, and debug.

Cortex-M55

The Cortex-M55 offers an easy way to implement AI for IoT with the ease-of-use of Cortex-M, a single tool chain, optimized software libraries, and an industry-leading embedded ecosystem.

The Cortex-M processor family

Cortex-M35P

Cortex-M35P is the first Armv8-M processor with tamper-resistance built in to bring physical security within reach of any IoT product. Cortex-M35P has the option of integrated digital signal processing (DSP), SIMD, and multiply-accumulate (MAC) instructions that simplify overall system design, software development, and debug.

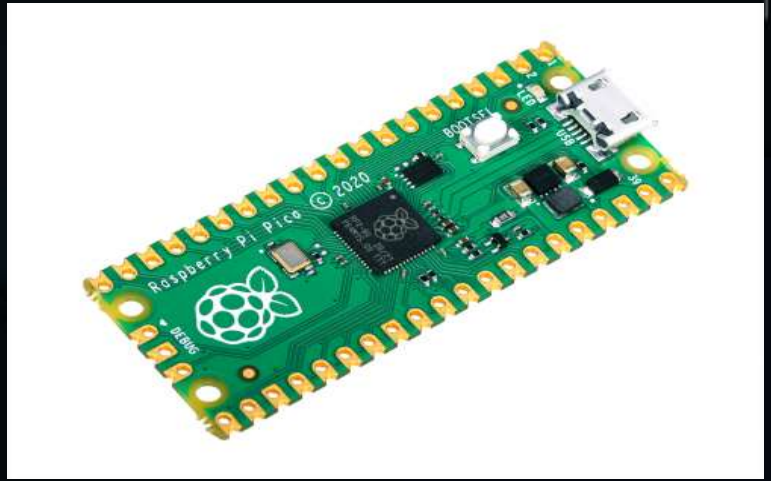
Cortex-M55

The Cortex-M55 offers an easy way to implement AI for IoT with the ease-of-use of Cortex-M, a single tool chain, optimized software libraries, and an industry-leading embedded ecosystem.

Raspberry Pi Pico

RP2040 microcontroller
Cortex M0+

C/C++
MicroPython
Assembly language



See
https://www.raspberrypi.com/products/raspberry-pi-pico/
https://www.raspberrypi.com/documentation/microcontrollers/rp2040.html
https://micropython.org/