Microprocessors

Fall 2020

1. Introduction to Class

Course Info: Instructor

Dr. Furkan Çaycı

- PhD - University of Delaware, USA
  - High performance platforms for beam projection and adaptive imaging applications
- Website: https://www.furkan.space/
- Office Hours: Message from MS. Teams
- Office Location: EMB 127
- E-mail: Use MS Teams to msg me.

Course Info: Teaching Assistants

- Mecit Emre DUMAN
- Erdem KÖSE
- Elif Betül ŞEN ÖZEN
- Selim Şahin

Use MS Teams to reach them.

Course Info: Links

- **You can find all the info on** https://micro.furkan.space/ . Please check there frequently.
- ELEC 334 - Syllabus https://micro.furkan.space/syllabus/
- ELEC 335 - Syllabus https://micro.furkan.space/elec335syllabus/
- We will use MS Teams for all the content management.
  - Find ELEC 334/335 group and sign up.

Course Info: Objective

Main objective of **Microprocessors** course is to **present** a **detailed** insight into microprocessors and get students **proficient** with common practices on **designing** microprocessor based systems.

Course Info: ELEC 334 Methodology

- We will have a lecture on Thursdays 9:30 - 12:30.
- The lectures will be recorded and uploaded for later viewing.
- Attendance will not be taken.
- There will probably a quiz in each lecture. We will talk about this in a bit.
- The lecture will probably take 1-2 hours. After the break, I am planning to do lab kinda sessions, but we will see how this goes.
- English / Türkçe

Course Info: ELEC 335 Methodology

- The labs will be assigned weekly, and you will work on the labs **individually**.
- The labs will be due on **Thursday nights**.
- There will be 8-10 labs throughout the semester.
- You will write a **report** and **record a video** of demonstration. We will talk about the specifics sometime later.

Course Info: Books and References

- We will use Yiu, J., 2015. **The Definitive Guide to ARM Cortex-M0 and Cortex-M0+ Processors**. as the main book, at least for the first couple of weeks.
- There will be a lot of reference and datasheet readings. (check website)
  - STM32G031K8 Datasheet
  - Nucleo-G031K8 User Manual
  - RM0444 - Reference Manual for STM32G0x1 Devices
  - PM0223 - Programming Manual for STM32G0
  - Cortex-M0+ Devices Generic User Guide
  - Cortex-M0+ Technical Reference Manual

Course Info: Hardware - General Info

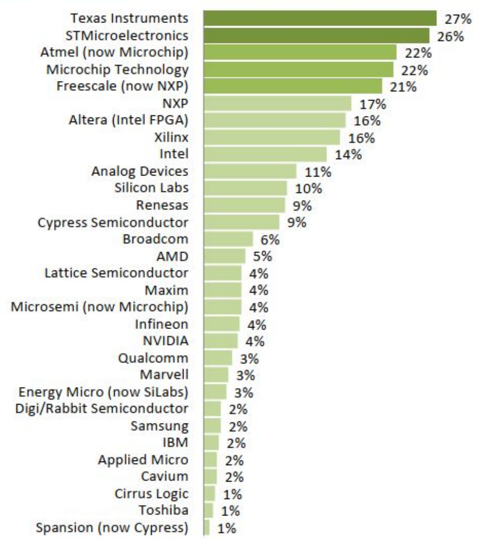
- I don't want you to fall behind on designing hardware systems, so there will be **a lot of hands on experiments**. For this purpose I selected a couple components and modules that we will try to use throughout the semester.
- Since it is almost impossible to work as a team for hardware projects, you will need to get all these **individually**.

Course Info: Hardware - Board

- We will go with ARM ecosystem since it is the most popular (and dominating) microprocessor line out there.
  - **ARM Cortex M0+**
- The main board for this course will be **Nucleo G031K8 board** from ST Microelectronics
- empa:::store should have a bunch of them in stock, and they will resupply if they run out of boards. You can buy with discount.

EETimes Embedded Markets Study 2019

Please select the processor vendors you are currently using.



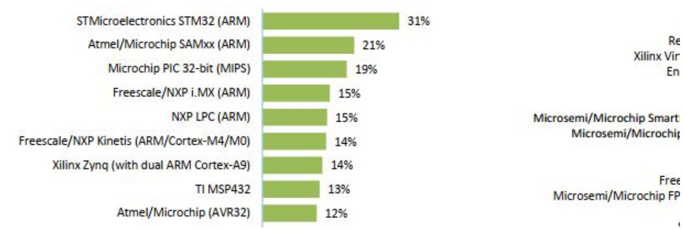
Merged Brands Combined	%
Microchip/Atmel/Microsemi (Net)	40
NXP/Freescale (Net)	28
Intel/Altera (Net)	26
Silicon Labs/Energy (Net)	10
Cypress/Spansion (Net)	9

Top Four Brands by Region:
Americas: TI, Microchip, STMicro, Atmel
EMEA: STMicro, NXP, TI, Atmel
APAC: TI, Atmel, Freescale, STMicro

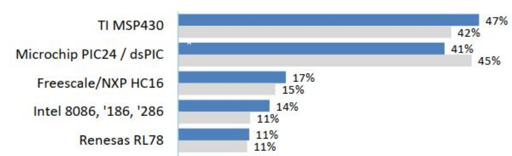
2019 (N = 458)

EETimes Embedded Markets Study 2019

Which of the following 32-bit chip families would you consider for your next embedded project?



Which of the following 16-bit chip families would you consider for your next embedded project?



Course Info: Hardware - Board

Figure 1. NUCLEO-G031K8 top view

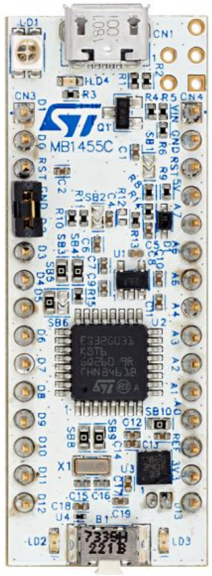


Figure 2. NUCLEO-G031K8 bottom view

Course Info: Hardware - Modules

- A Breadboard.
- Wiring, mostly M-M
- 2 x LEDs, 2 x Push Buttons (Tactile switches)
- Resistors and Capacitors of various sizes (2 of each at least). Example list (10, 390, 470, 1k, 10k, 50k, 100k, 220u, 100u, 10u, 1u, 100n, 47n)
- 2 x 10k Pot
- 4x Seven Segment Display (common anode / cathode does not matter)
- 4x4 Membrane Keypad
- Analog microphone w/ amp.
  - If you find no amp mic, you can build an amplifier circuit.
- 2 x 24LC512 I2C EEPROM example
- MPU6050 6DOF IMU example
- Speaker
- LM386 Lower Power Amp
- a couple MOSFETs and BJTs (ie. 2N5551, BC237, BS170)

Get them wherever you can.

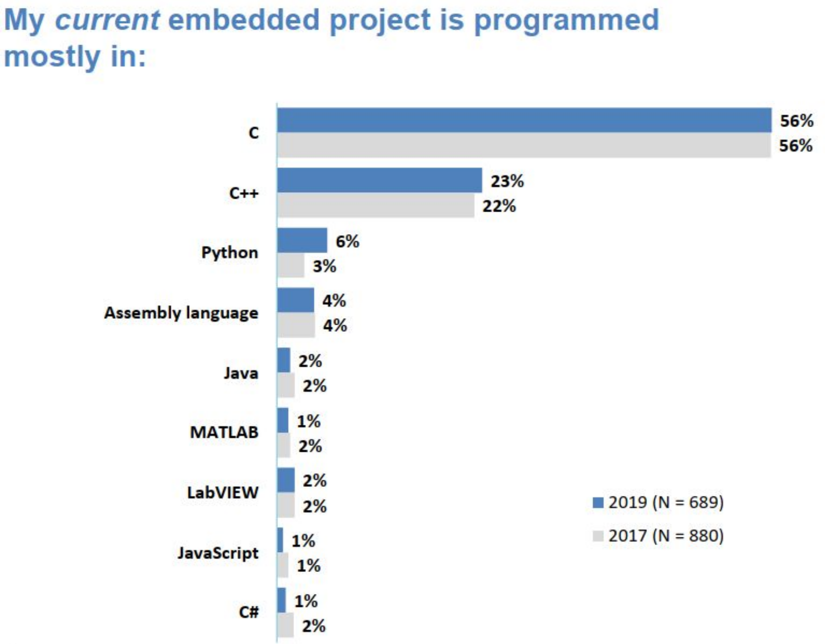
Course Info: Software

- We will use **STM32CubeIDE** which is an Eclipse based IDE and has a debugger as well.
  - Windows / Linux / macOS
  - There are many other alternatives (and more popular ones as well) such as Keil, IAR, Segger, vs. but most of them have limitations.
- There is an ARM Assembly emulator called VisUAL that you can play with to visualize instruction/register/memory usage. We will play with this for the first 2-3 weeks until the boards arrive.

Course Info: Programming Language

- We will use **C** because it is *the de facto standard* for embedded systems.
  - Might change soon. C++ is popular, Python is getting popular, Rust is incoming...
  - **Double edged-sword. Precise** control over what the processor is doing, but **dangerous** in case of a mistake.
    - i.e no compile-time error catching or memory safety.
  - **Predictable** behavior.
    - i.e no garbage collection or preemption
- We will start with **Assembly**
  - The compiler translates C into assembly, and does a really good job, but understanding Assembly will help you greatly in the case of problems
  - Rarely you might need to write assembly, but rarely...

EETimes Embedded Markets Study 2019



Course Info: Tentative Weekly Schedule

- Week x1 - Introduction to Course
- Week x2 - Architecture
- Week x3 - Assembly Language Introduction
- Week x4 - Assembly Language Memory and Control
- Week x5 - Embedded C, Toolchain and Debugging
- Week x6 - Interrupts
- Week x7 - Timers
- Week x8 - Modulation
- Week x9 - Serial Communications I
- Week xA - Serial Communications II
- Week xB - Analog Interfacing
- Week xC - Memory and DMA
- Week xD - RTOS
- Week xE - Wireless Communications

Course Info: Exams

- Due to remote learning, there will be **no midterm** (for sure) and **no final** (possibly) for now.
- **However**, the university might decide to hold the finals face–to–face, which will introduce a final, and this final will cover all the material that you have seen in the course.
- So, consider **both possibilities and plan accordingly**.

Course Info: Assignments

- Homework
  - Several assignments throughout the semester.
  - You may and should discuss the assignments with your study partners, but you are required to complete your assignments individually.
- Projects
  - There will be 3 projects for the ELEC 334 course.
  - Projects will be **individual** and will be built and demonstrated by you. Details later.

Course Info: Projects and Labs

- Since there is no exams, **we will do 3 projects for ELEC 334** course.
- Each project will be a little **challenging** but doable. Expect to spend a good amount of time.
- Projects will be **individual** and will be built and **demonstrated by you**.
- More details later.
- ELEC 335 will be based on labs which are again individual work.
  - There will be a lab each week except the weeks that the projects are due.

Course Info: Lab Reports I

- For ELEC335 - Lab reports will follow ELM235 style.
- **What you should do:**
  - Write a one-paragraph introduction about the content.
  - Add pictures of your setup as **figures** and number them.
  - Add your **code** as appendix formatted nicely.
  - Comment on how things work (or do not work), what you expected, and why you expect it.
  - Attach relative sections from datasheets as appendix.
  - all these **on your own words.**

Course Info: Lab Reports II

- For ELEC335 - Lab reports will follow ELM235 style.
- **What you should NOT do:**
  - Copy-paste information from the web.
    - This is not your own words, and you are stealing someone else's work and presenting as it is yours.
  - Write the comments saying, *it worked as expected.*
    - I already know how it should work. I am trying to understand what you understood from the lab.
  - Write empty sentences just to fill the report.
    - Writing two pages about a simple bounce mechanism that you implemented is wasteful.

Course Info: Grading

- Following is the items that will contribute to your final grade. **Make sure to understand the meaning of these percentages to avoid any surprises.**
- For w/o final option:
  - 10% - Quiz
  - 25% - Homework
  - 65% - Project (15 / 20 / 30)
- For w/ final option:
  - 7% - Quiz
  - 20% - Homework
  - 53% - Project (12 / 16 / 25)
  - 20% - Final
- Your final letter grade will be based on class standings, but try not to fall below 50.

Course Info: Academic Dishonesty

- As a principle, **you should not present someone else's work as yours** (code, report, drawings, etc.). It might be for getting a higher grade, or simply saving the day, but remember, a lot of your friends and peers are working hard to get that grade. This is disrespectful for both you, your friends, us and that someone else that you stole the material from.
- We do have means of comparing codes and finding replicas (even if you make changes to the code), so please do not go that route, and avoid any unpleasant consequences.
- This semester I will change my approach. **If I catch a copy or a clone work, you will automatically fail from the course.** There will be no second chances.

Let's talk about what we learned last semester...

Review: 5 Great Ideas in Comp Arch

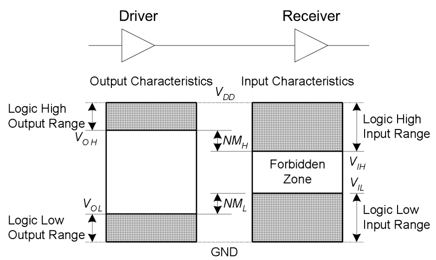
- **Moore's Law** - Designing through trends.
- **Abstraction** - Layers of representation / interpretation
- **Principle of Locality** - Memory Hierarchy
- **Parallelism** - Better performance
- **Dependability via redundancy** - Reliability

Review: Representation of Numbers

- Number Systems
  - Base-2 (binary)
  - Base-10 (decimal)
  - Base-16 (hexadecimal)
- 2's complement
- Sign extension
- BCD / Gray / ASCII codes

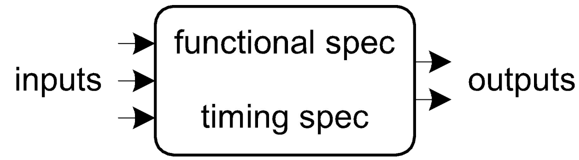
Review: Logic families and Noise margins

Logic Family	V_{DD}	V_{IL}	V_{IH}	V_{OL}	V_{OH}
TTL	5 (4.75 - 5.25)	0.8	2.0	0.4	2.4
CMOS	5 (4.5 - 6)	1.35	3.15	0.33	3.84
LVTTL	3.3 (3 - 3.6)	0.8	2.0	0.4	2.4
LVCMOS	3.3 (3 - 3.6)	0.9	1.8	0.36	2.7



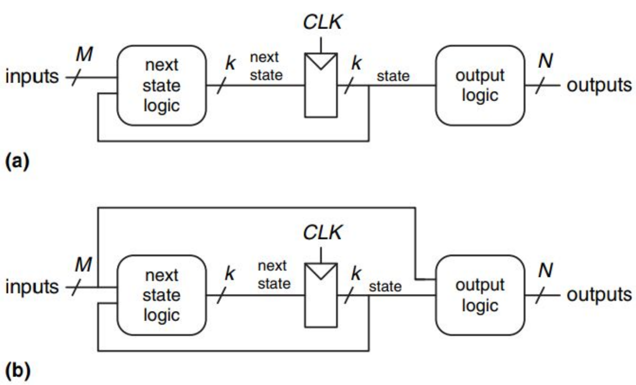
Review: Combinational & Sequential Logic

- **Combinational Logic**
    - Memoryless
    - Outputs determined by current values of inputs
- **Sequential Logic**
    - Has memory
    - Outputs determined by previous and current values of inputs



Review: Finite State Machines

- Two types of finite state machines differ in output logic:
    - **Moore FSM:** outputs depend only on current state
    - **Mealy FSM:** outputs depend on current state and inputs



Review: Architecture vs. Microarchitecture

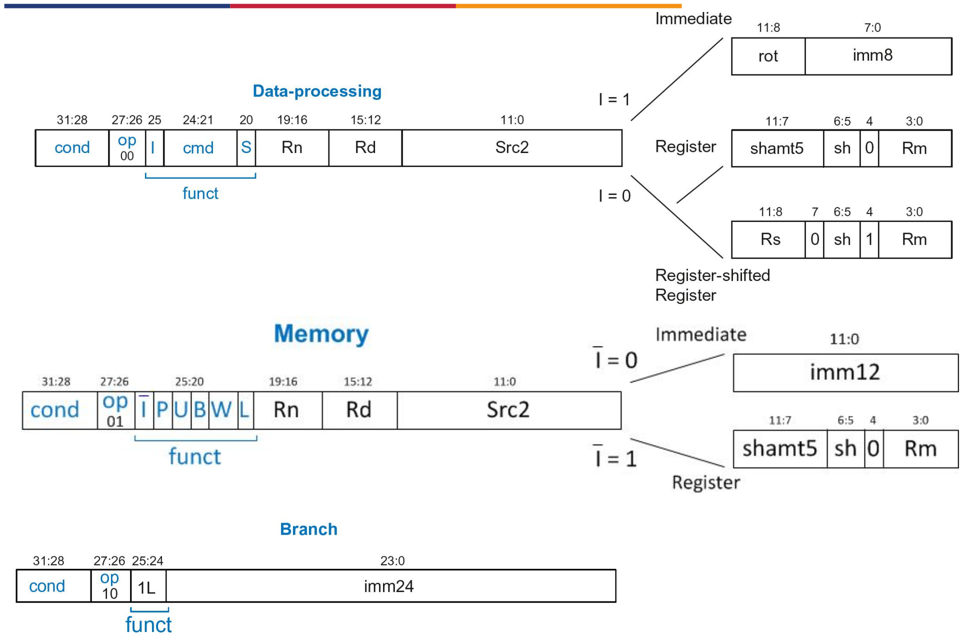
Architecture (Instruction Set Architecture)

- Programmer's view of computer.
- Defined by instructions & operand locations

Microarchitecture (Organization):

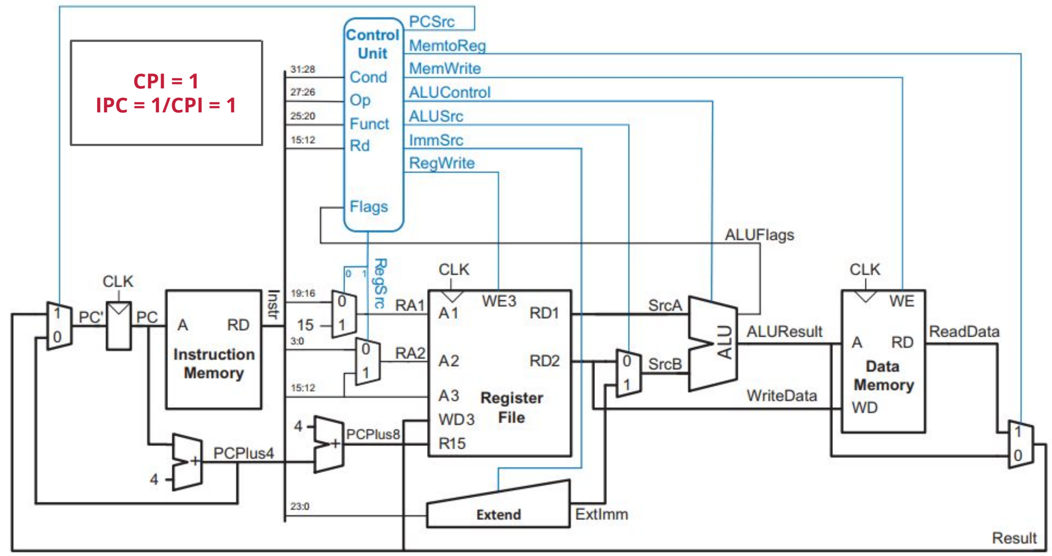
- How to implement an architecture in hardware
- Connection between logic and ISA
- Specific arrangements of registers, ALUs, FSMs, memories and other logic building blocks
- Different trade-offs of performance, cost and complexity

Review: Instruction Formats



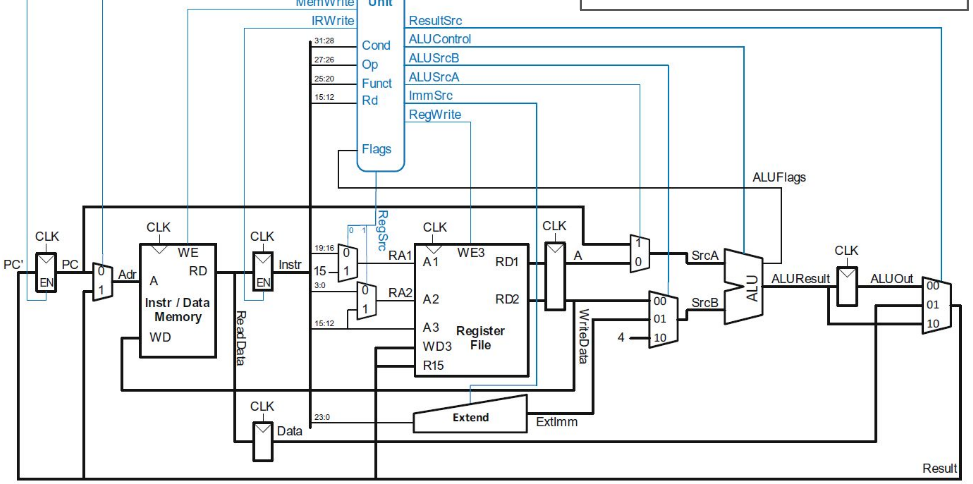
Review: Single-Cycle ARM Processor

CPI = 1
IPC = 1/CPI = 1



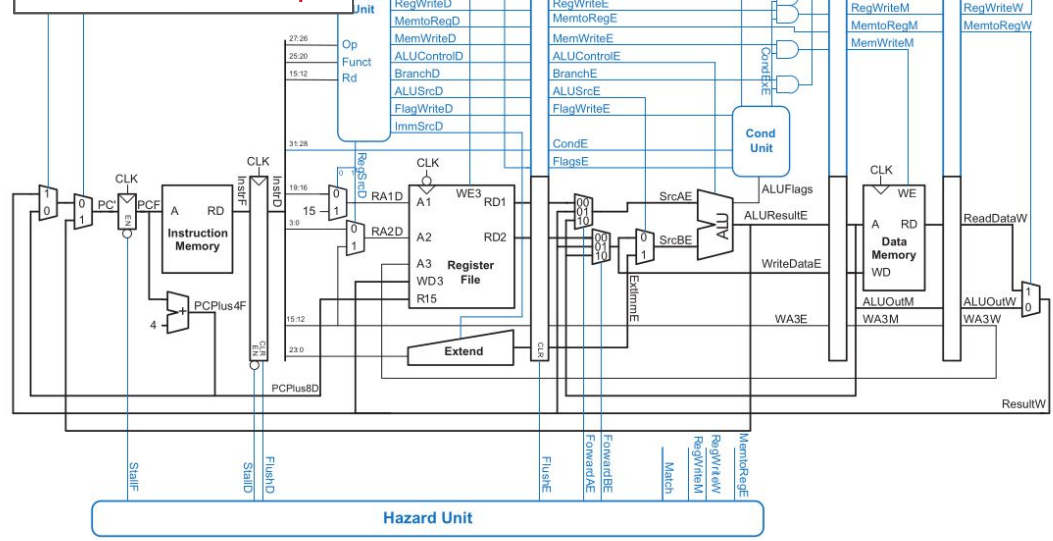
Review: Multicycle ARM Processor

CPI > 1
- **Depends on instruction types**
- **3≤CPI≤5 for this example**



Review: Pipelined ARM Processor

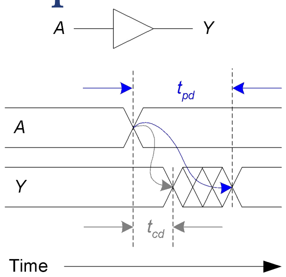
CPI ≥ 1
- **Depends on instruction types and inter-instruction dependencies**
- **1≤CPI≤3 for this example**



Transition from Logic Design to Microprocessors

For Combinational Logic:

- **Propagation delay:** t_{pd} = max delay from input to output
- **Contamination delay:** t_{cd} = min delay from input to output



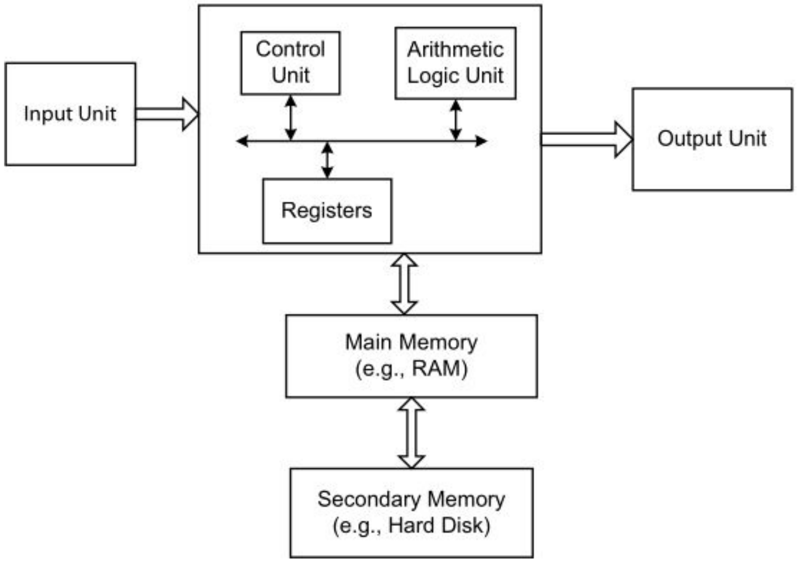
- In *Logic Design course*, we cared about **timing within a clock cycle** and tried to **optimize** what we can achieve **within that clock cycle**.
- In *Microprocessors course*, we don't care about single clock cycle, but we care about **how fast** we can **compute** and try to **optimize number of clock cycles**.

Let's talk about computers...

Classes of Computers

- **Desktop computers**
  - General purpose, variety of software
  - Subject to cost, performance tradeoff
- **Server computers**
  - Network based
  - High capacity, performance, reliability
  - Range from small servers to building sized
- **Embedded computers**
  - Hidden as components of systems
  - Strict power, performance, cost constraints

General Purpose Computer



Embedded Computers

- Computing systems **embedded within** electronic devices. (*Embedded Systems*)
- Repeatedly carry out a **particular** function or **a set of** functions.
- Contain **both hardware and software**
- **Billions** of units produced yearly, versus **millions** of desktop units.



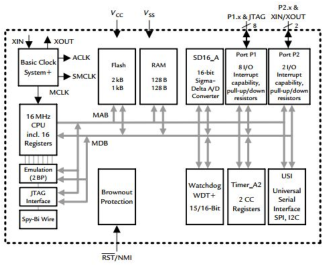

Common characteristics of embedded systems

- **single-functioned** - executes single program
  - repeated execution, might implement sophisticated algorithms
- **real-time operation** - must finish operations by deadlines
  - **hard deadline** - missing deadline causes failure
  - **soft deadline** - missing deadline causes degraded performance
- **tightly-constrained** - low cost, low power, fast, small, ...
  - limited memory, battery-operation, ...
  - non-functional requirements

Embedded systems are everywhere...

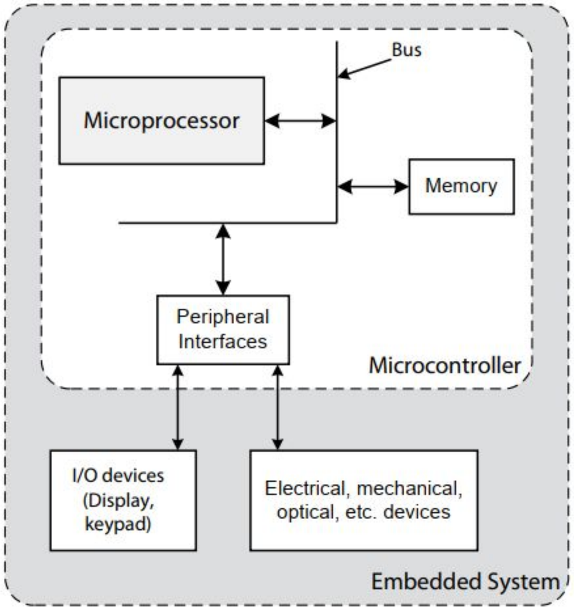
- **Communication**: Radios, telephones, cellular phones, answering machines, fax machines, wireless routers.
- **Consumer electronics:** Washing machine, clocks and watches, games and toys, remote controls, audio/video electronics.
- **Automotive systems:** Braking system, electronic ignition, locks, power windows and seats, collision avoidance.
- **Commercial usage:** ATM machines, bar code readers, elevator controllers.
- **Medical treatments:** Cancer treatments, dialysis machines, blood pressure measuring equipment, electrocardiography
- **Industrial:** Process automation, oil refineries, food processing plants, paper and board mills, etc.
- **Military:** Missile guidance systems, global positioning systems, surveillance systems

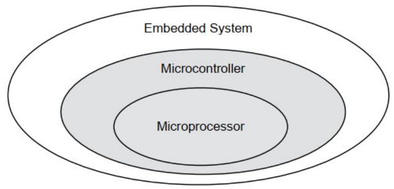
What are the differences between these?



- Different requirements:
  - Power consumption
  - Speed of execution
  - System size and weight
  - Performance accuracy
- Which leads to the choice of a ***microprocessor***
  - Processing rate, processor size
  - Different types of I/O devices available
  - Memory size (both ROM and RAM)

A generic view of an Embedded System





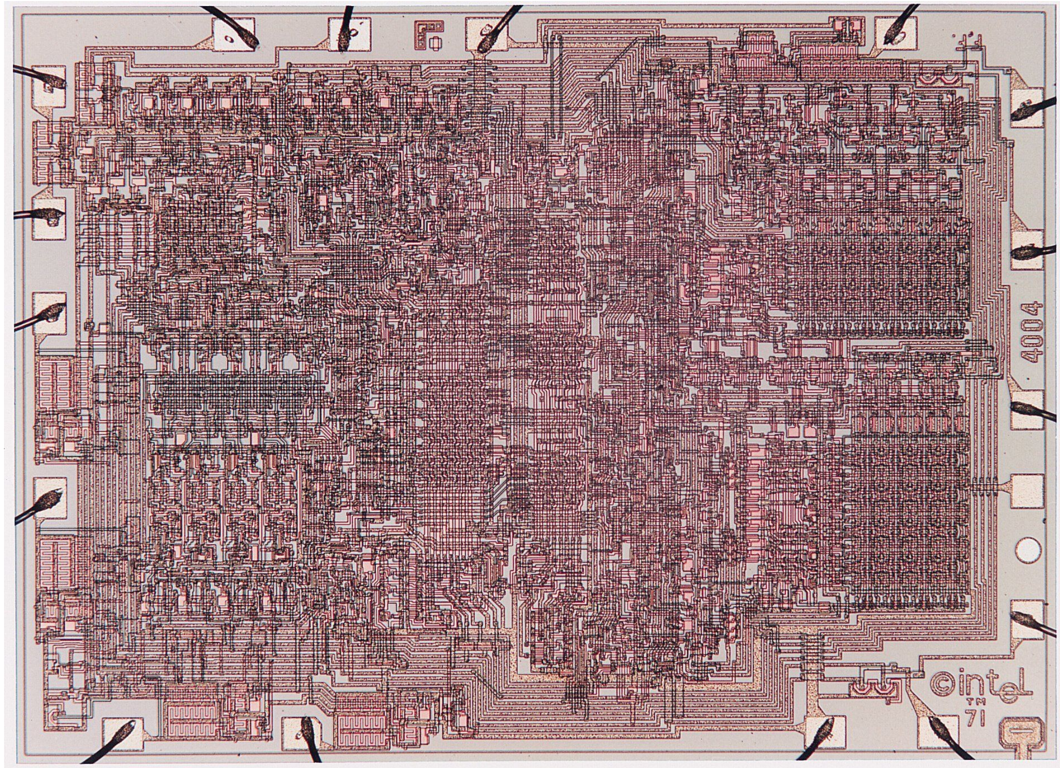
Why microprocessors?

Alternatives include:

- **Digital Signal Processors** - microprocessors optimized for signal processing
- **Graphics Processing Units** - power hungry for small applications, mostly for high-performance computing
- **Field Programmable Gate Arrays** - holy grail, slow development, vendor specific tooling
- **Application Specific Integrated Circuits** - more specialized, high cost, little to none scalability, no re-programmability
- Microprocessors often offer efficiency, same libraries can be used across different iterations, high scalability, programmability, heavily pipelined, built-in power consumption optimizations
  - e.g. **Cortex-M0+** consumes ~0.27 μA Standby mode

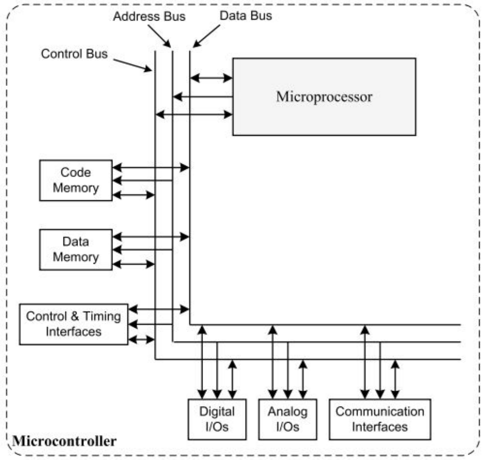
Early History...

- around 1940s, a computer that is designed for aircraft simulation that supports real-time operation: **MIT Whirlwind**, a lot of vacuum tubes
- in 1970, first microprocessor: **Intel 4004**, 2250 transistors, 750 kHz frequency
- in 1972, first handheld calculator using several chips: **HP-35**
- in 1980s, automobile industry started using microprocessors: engine timing, fuel efficiency, emissions, cruise, ...



Microcontroller

- A **microcontroller** combines a microprocessor along with *RAM*, *ROM* and peripherals (*IO*) in an Integrated Circuit (IC).
- Everything is connected through a bus (or a couple busses).
- Usually there will be many selections based on the same microprocessor.



What is inside a Microcontroller?

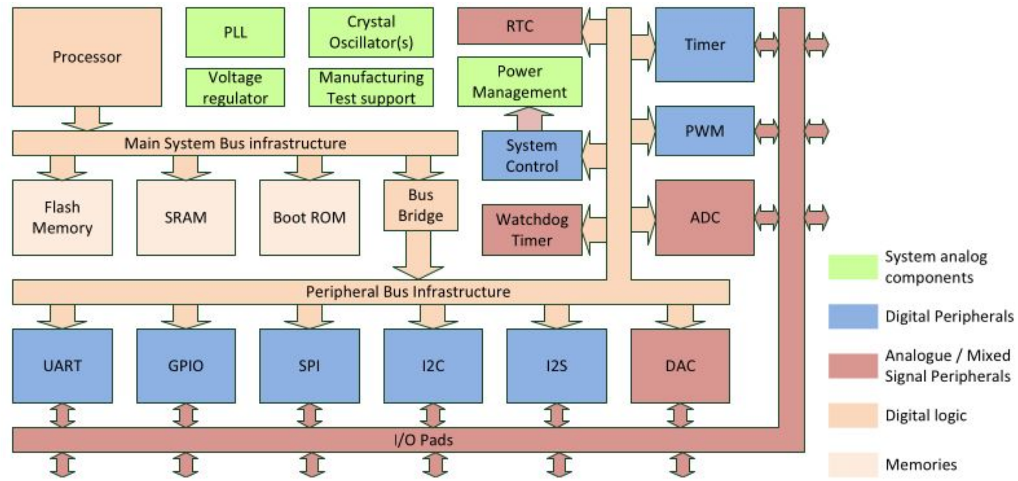


Figure 1.9
A simple microcontroller.

Characteristics of microprocessors

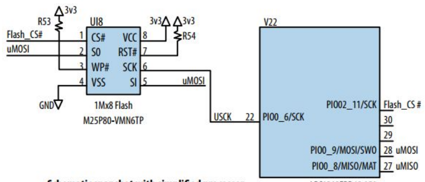
- Low power
- Fast interrupt response
- High code density
- Debug
- OS support
- Ease of use
- High software portability and reusability
- Upgrade and downgrade path
- Tool chain support
- Low cost

Design Methodologies

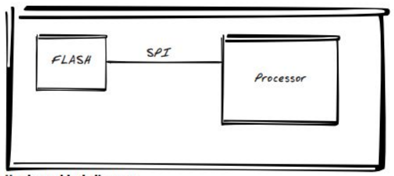
- a procedure for designing a system
- good methodology helps you avoid mistakes and gives you a clear representation of the system in all abstraction levels
- software engineering tools, computer-aided design (CAD) tools, ... can be used to keep track of changes and automate some tasks
- **block diagrams, flow charts**, UML (Unified Modeling Language)

Design Methodologies - Block diagram

Start your design by **drawing a box for the processor** and other boxes for connecting the **peripherals** and **external components**.



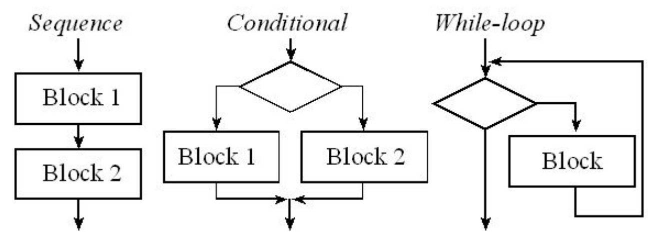
Schematic snapshot with simplified processor

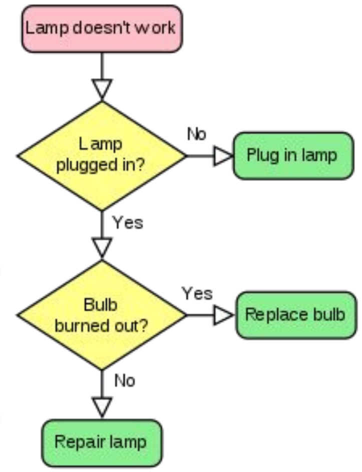


Hardware block diagram

Design Methodologies - Flow Chart

- After your hardware diagram is complete, start by creating **boxes** for your software functions, and lay out your algorithm.
- Use a **flowchart** to describe your software operation.



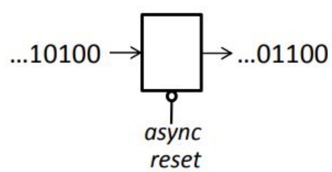


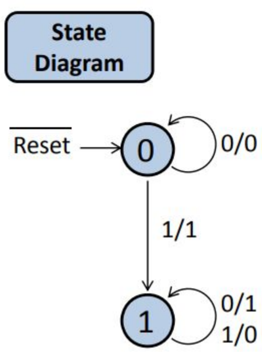
Example: Serial 2's complementer

Design a serial 2's complementer using Mealy machine.

Two states:

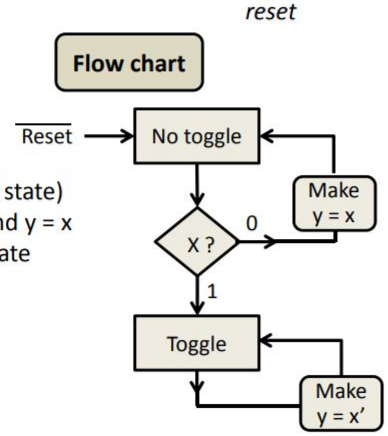
- Until first 1 occurs, `output = input`
- When first 1 occurs, `output = !input`



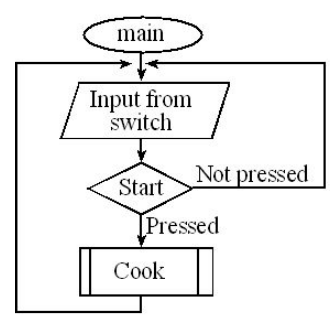


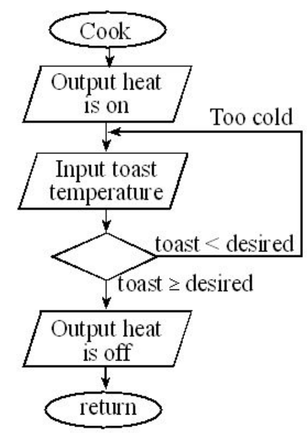
Algorithm

- No toggle state (initial state)
  If x = 0, stay in state and y = x
  If x = 1, go to toggle state
- Toggle state
  Stay in state and y = x'



Example: Toaster oven





An important closing note

We will work with ARM ecosystem, and learn to program a specific architecture with specific registers using specific tools...

But ... the technology will change!...

It is important for you to **think critically**, and **understand the principles behind decisions**. The rest is applying these principles to different technologies...

One of the important questions in this course is **How does this work?**

This week

- Install software
  - STM32CubeIDE + STM32G0 package
  - VisUAL
- Find / order hardware components
- Read Chapter 1 from Yiu
- Assignment and Reading

Links

- https://www.furkan.space/course/elec334/syllabus/
- https://www.furkan.space/course/elec334/elec335syllabus/
- https://www.embedded.com/wp-content/uploads/2019/11/EETimes_Embedded_2019_Embedded_Markets_Study.pdf
- https://www.direnc.net/tekli-breadboard
- https://salmanarif.bitbucket.io/visual/index.html
- https://spectrum.ieee.org/tech-history/silicon-revolution/chip-hall-of-fame-intel-4004-microprocessor
- http://uvicrec.blogspot.com/2011/09/understanding-intel-4004.html
- https://en.wikipedia.org/wiki/Flowchart