NANYANG TECHNOLOGICAL UNIVERSITY

School of Mechanical & Aerospace Engineering

Design, Machine, Control and Intelligence

MA4832

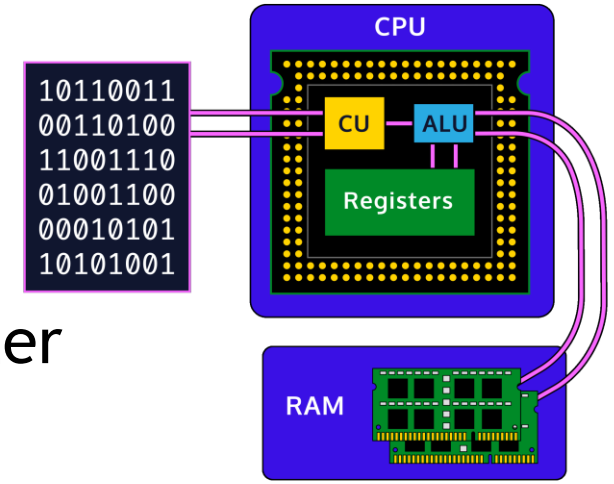
Microprocessor Systems

Xie Ming, PhD (France)

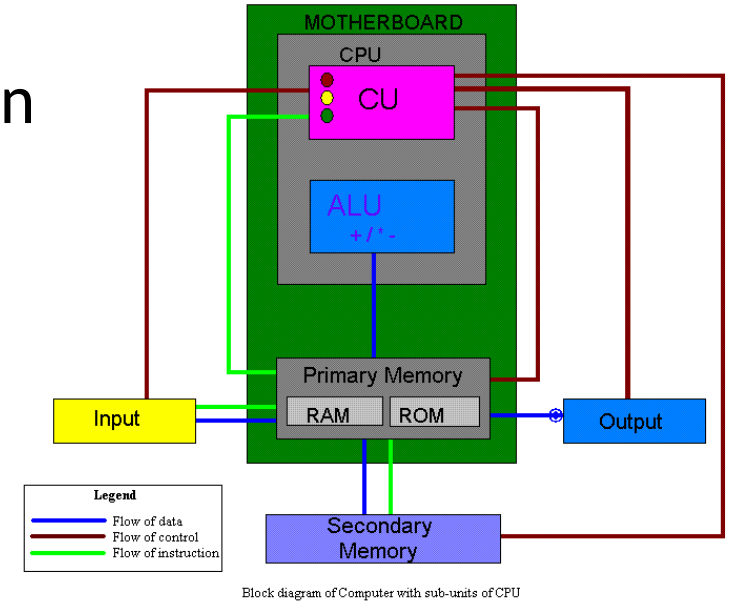
mmxie@ntu.edu.sg

http://personal.ntu.edu.sg/mmxie

Outline



- Lecture 1: Basics of ARM Microcontroller
- Lecture 2: ARM's Memories
- Lecture 3: ARM's Data Representation
- Lecture 4: ARM's Programming
- Lecture 5: ARM's Data Input/Output
- Lecture 6: ARM's Data Processing



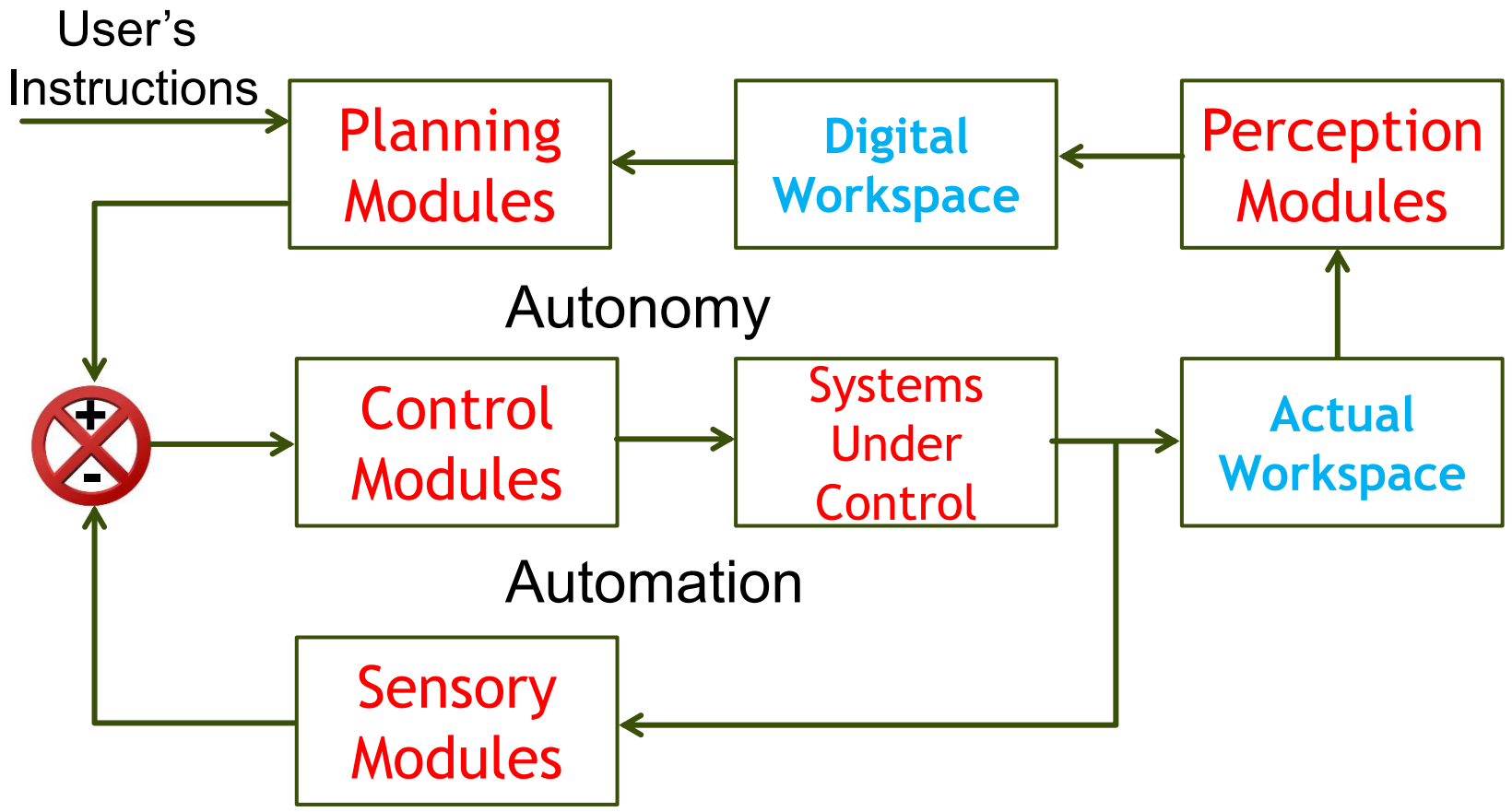
Block diagram of Computer with sub-units of CPU

Remember your mission as MAE undergraduates …

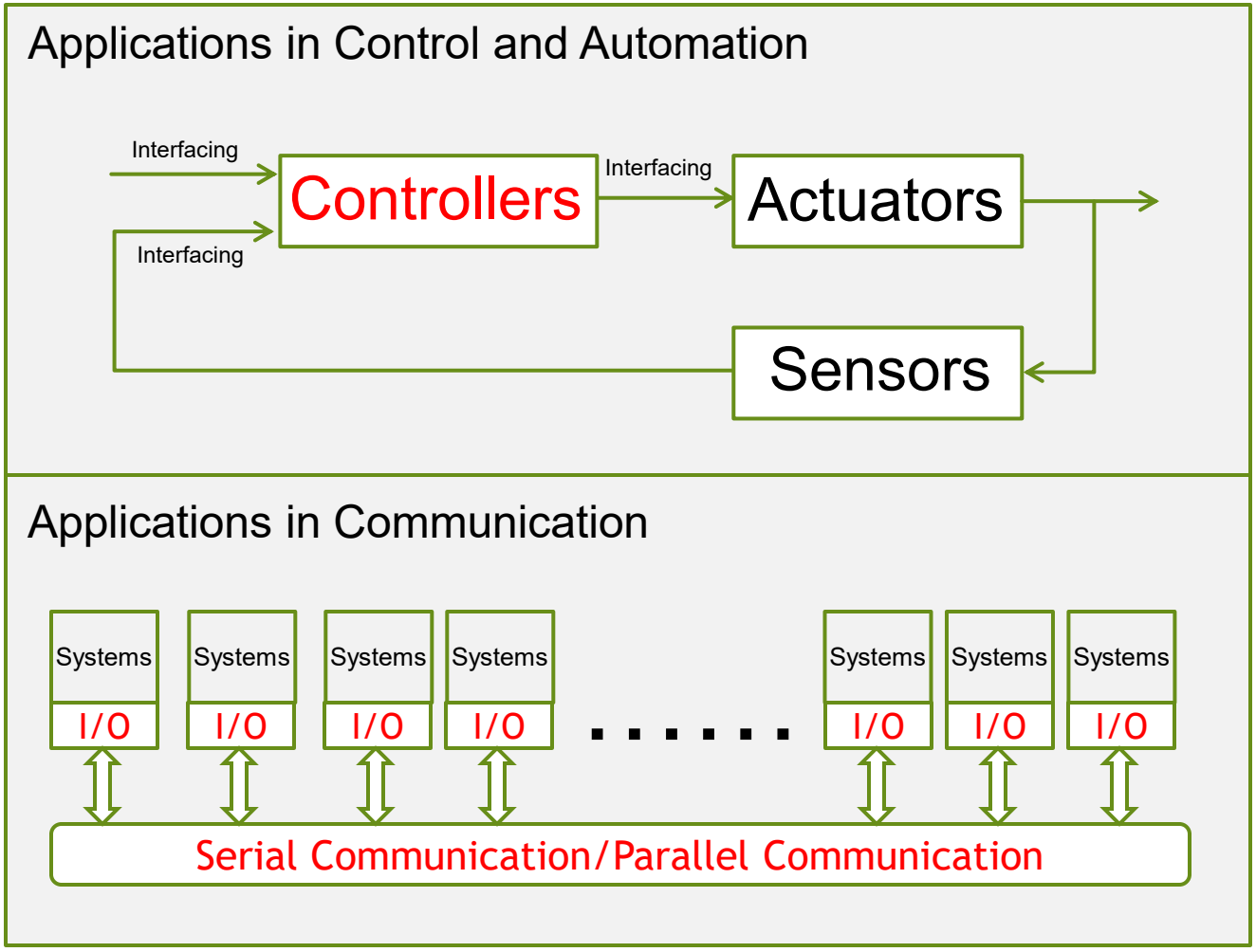
- You are here to grow your knowledge and skills so as to be able to design machines with controllable behaviors and hopefully in some intelligent ways.

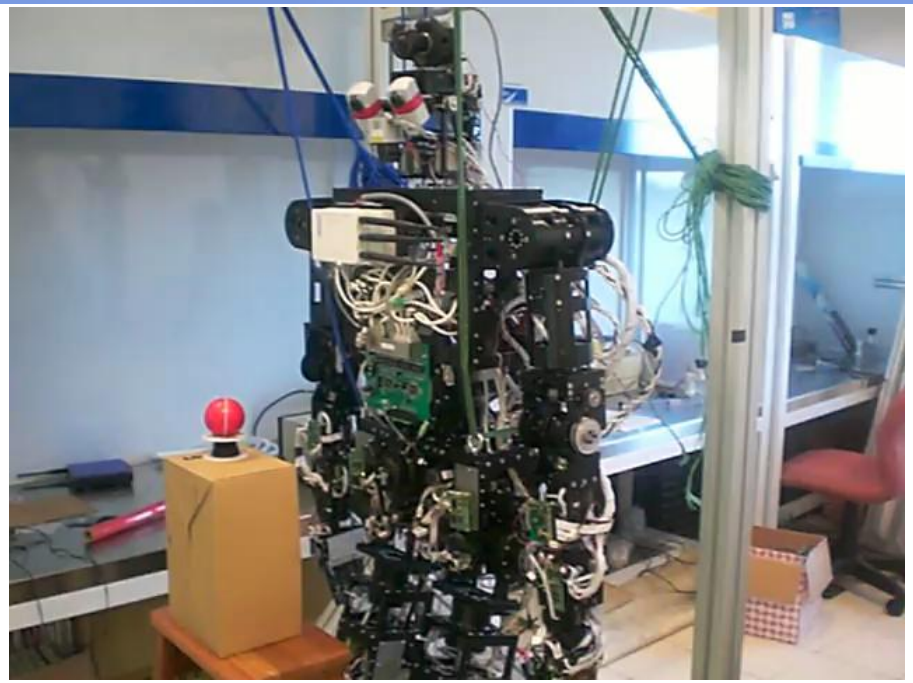
How to fulfill your mission?

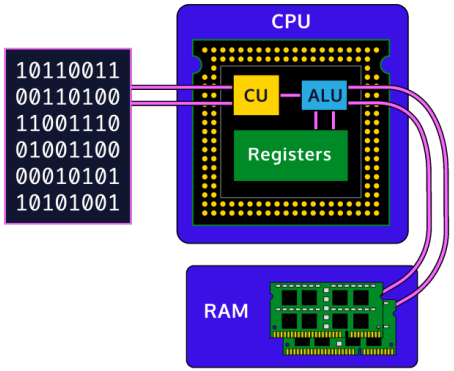
- To apply learnt knowledge and skills into the implementation of the following universal blueprint underlying all the intelligent machines or systems.

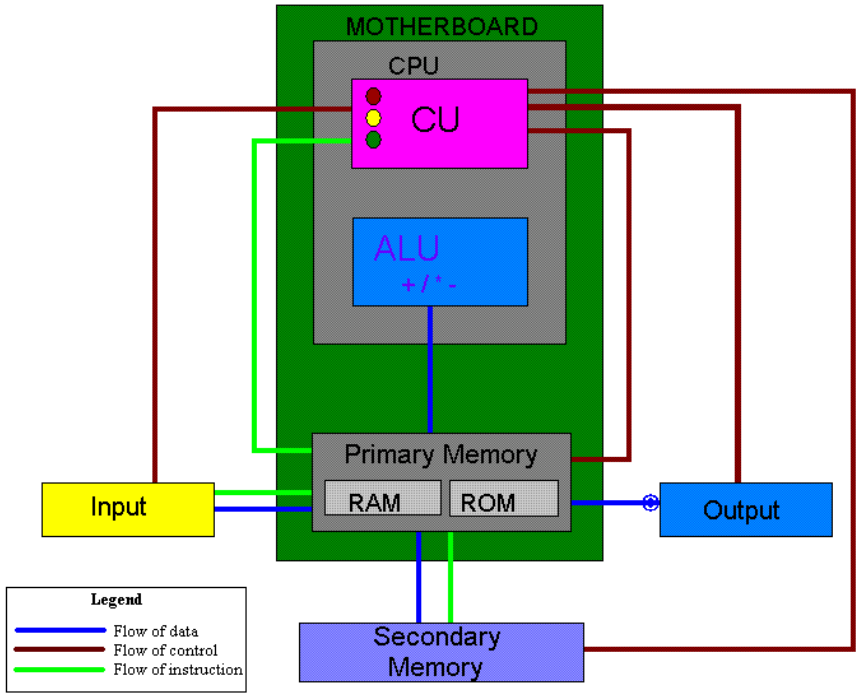


Why to study?


Applications in Control and Automation
Interfacing
Interfacing
Controllers
Interfacing
Actuators
Sensors
Applications in Communication
Systems
I/O
Systems
I/O
Systems
I/O
Systems
I/O
Systems
I/O
Systems
I/O
Systems
I/O
Serial Communication/Parallel Communication



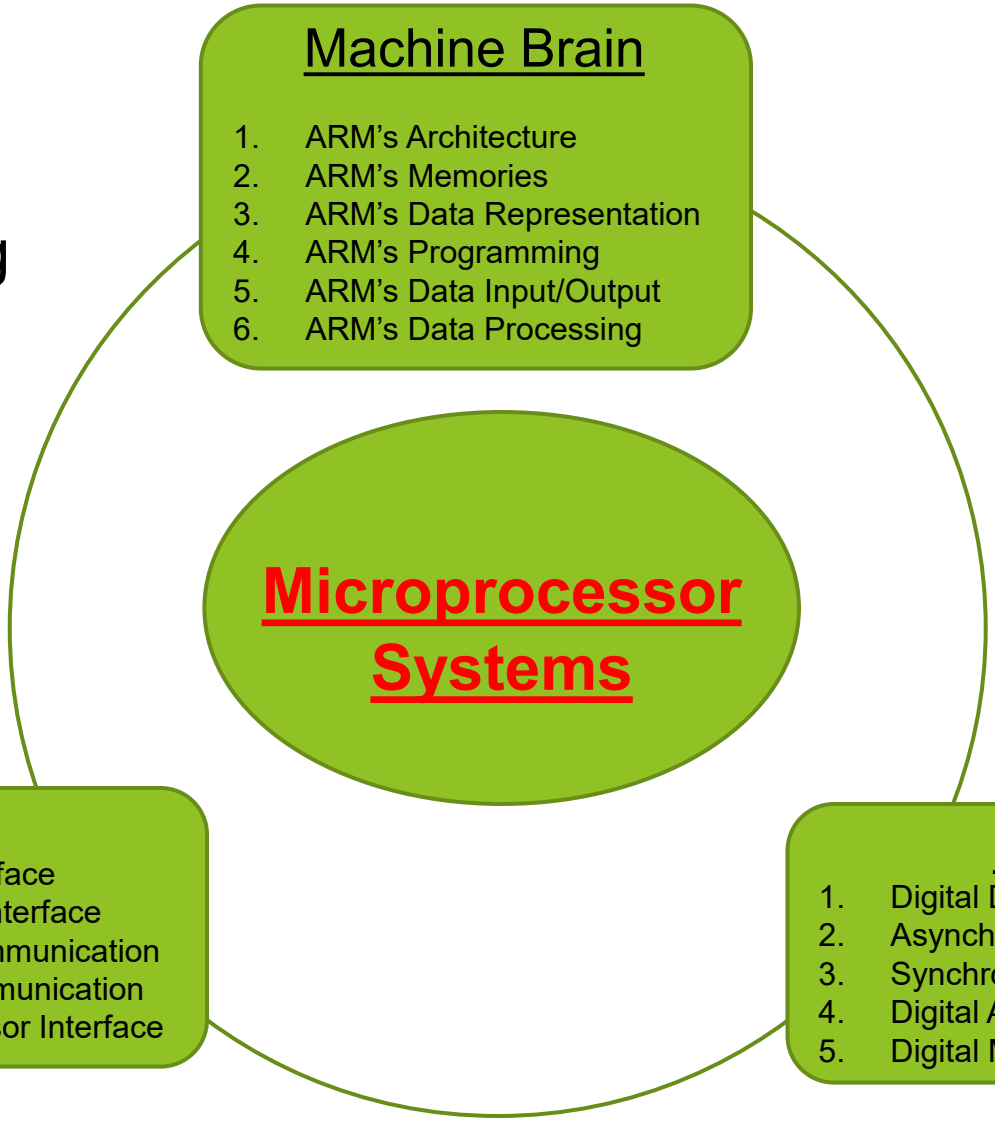




Block diagram of Computer with sub-units of CPU

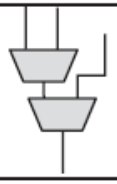
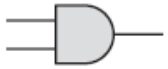
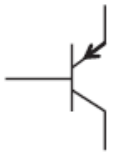
What to learn?

- Programming
- Interfacing



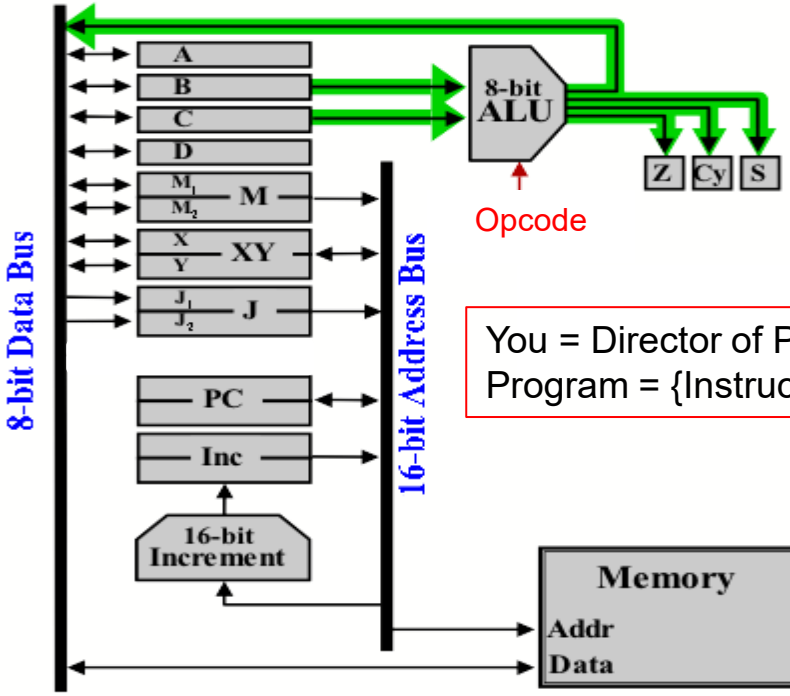
What is your role?

- Data = {Values, Symbols, Addresses, Instructions}
- Instructions = {Op Code + Addresses + Value/Symbol}

Applications/OS	
Languages	C++, Java
ISA	EOR r3,r2,r1 BEQ Table
Microarchitecture	
Gates	
Transistors	

Algorithms or Solutions

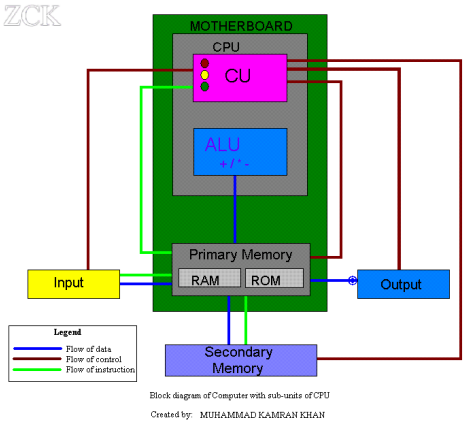
Problems to be solved



You = Director of Program
Program = {Instructions}

Memory = {Address + Data/Instruction}

How to learn?



- To understand data flows inside a microcontroller.
- To translate your solutions into data flows.
- To pay attention to <memory address> and <memory data>.
  - Memory Address: Address Label/Name and Address Value.
  - Memory Data: Data Label/Name and Data Value.

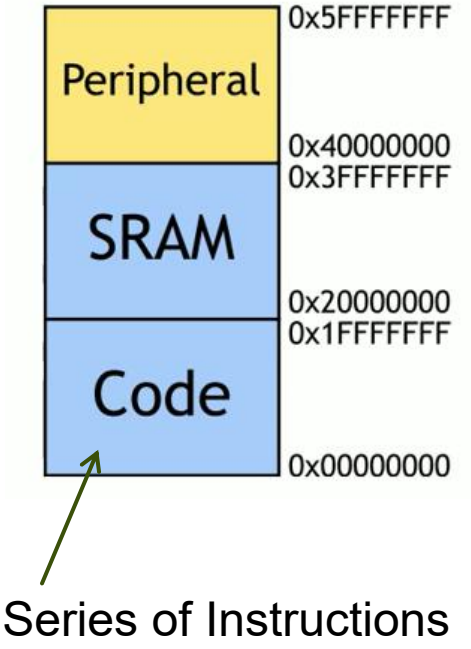
Memory Content

Data	Memory Address

- To pay attention to <memory address> and <memory code>.
  - Memory Address: Address Label/Name and Address Value.
  - Memory Code: Code Label/Name and Code Value.

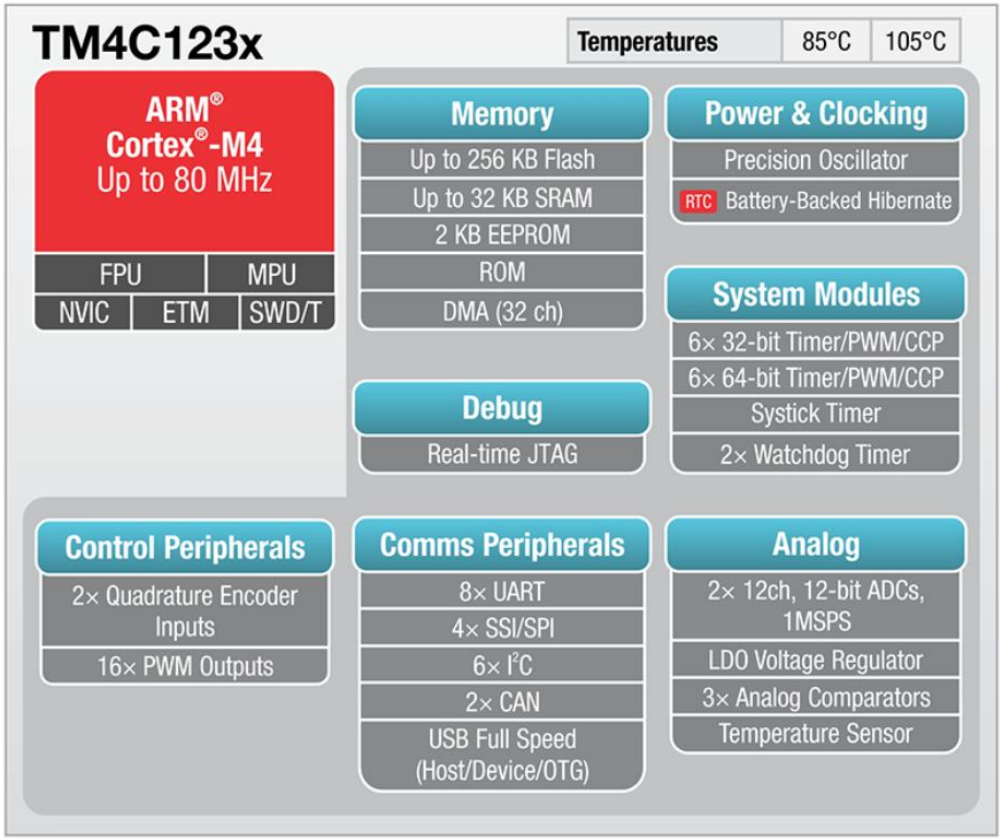
Memory Content

Instruction	Memory Address



Example of Using I/O Modules

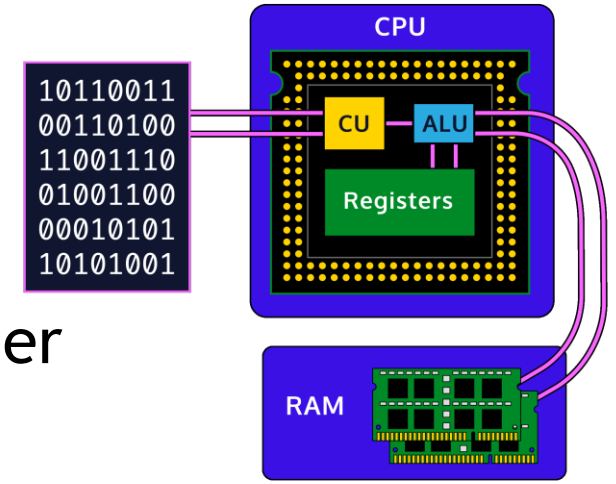
- Configure Control Registers
- Clear/Monitor Status Registers
- Read/Write Data Registers
- Instructions:
  - MOV <address of destination>, <source of value>
  - LDR <address of destination>, <source of value>
  - STR <source of value>, <address of destination>



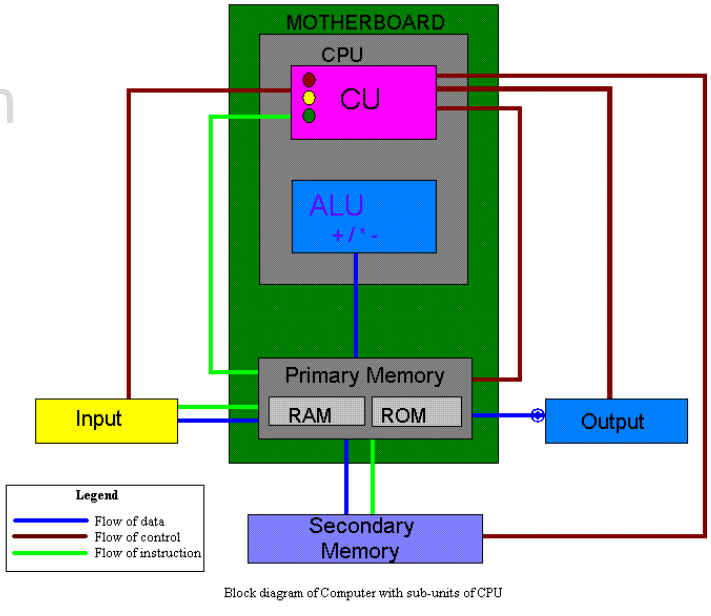

```
MOV   R0, #0x11
MOV   R1, #2560
MVN   R2, #4
MOVW  R3, #0xC0DE
MOVT  R3, #0xFEED
MOV   R4, R1
```

Word = 2 Bytes

Today's Lecture …



- Lecture 1: Basics of ARM Microcontroller
- Lecture 2: ARM's Memories
- Lecture 3: ARM's Data Representation
- Lecture 4: ARM's Programming
- Lecture 5: ARM's Data Input/Output
- Lecture 6: ARM's Data Processing



Block diagram of Computer with sub-units of CPU

NANYANG TECHNOLOGICAL UNIVERSITY

School of Mechanical & Aerospace Engineering

Design, Machine, Control and Intelligence

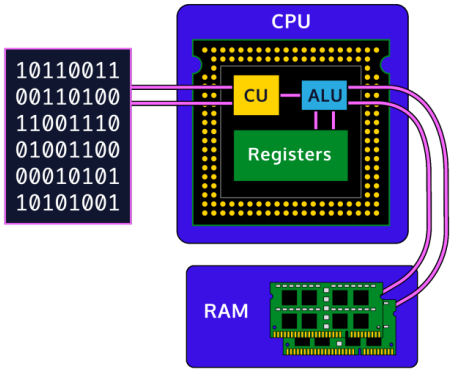
MA4832

Basics of ARM Microcontroller

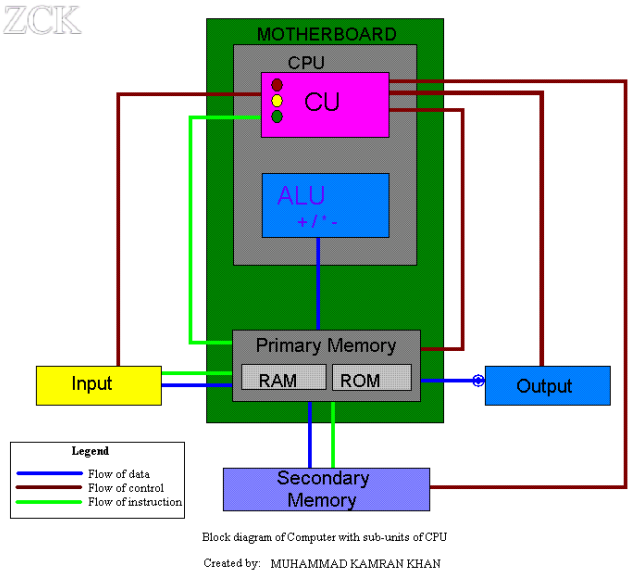
Xie Ming, PhD (France)

http://personal.ntu.edu.sg/mmxie

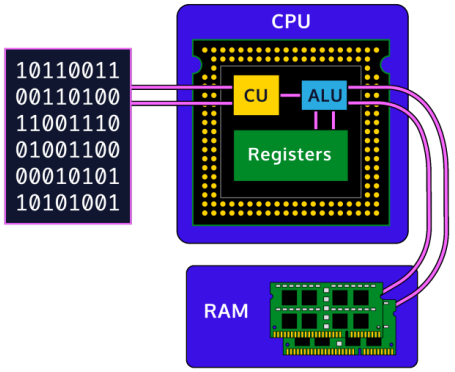
Outline



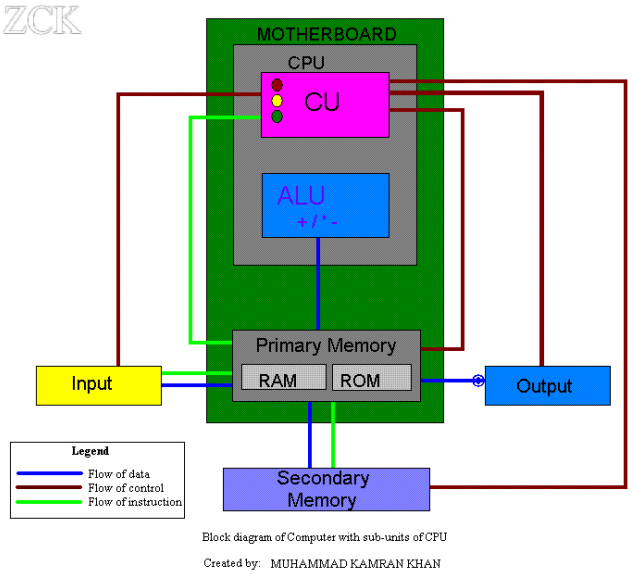
- Basics of Brains
- Inventors of Digital Computers
- Family of ARM Microprocessors
- Integrated Development Environment



Outline



- Basics of Brains
- Inventors of Digital Computers
- Family of ARM Microprocessors
- Integrated Development Environment

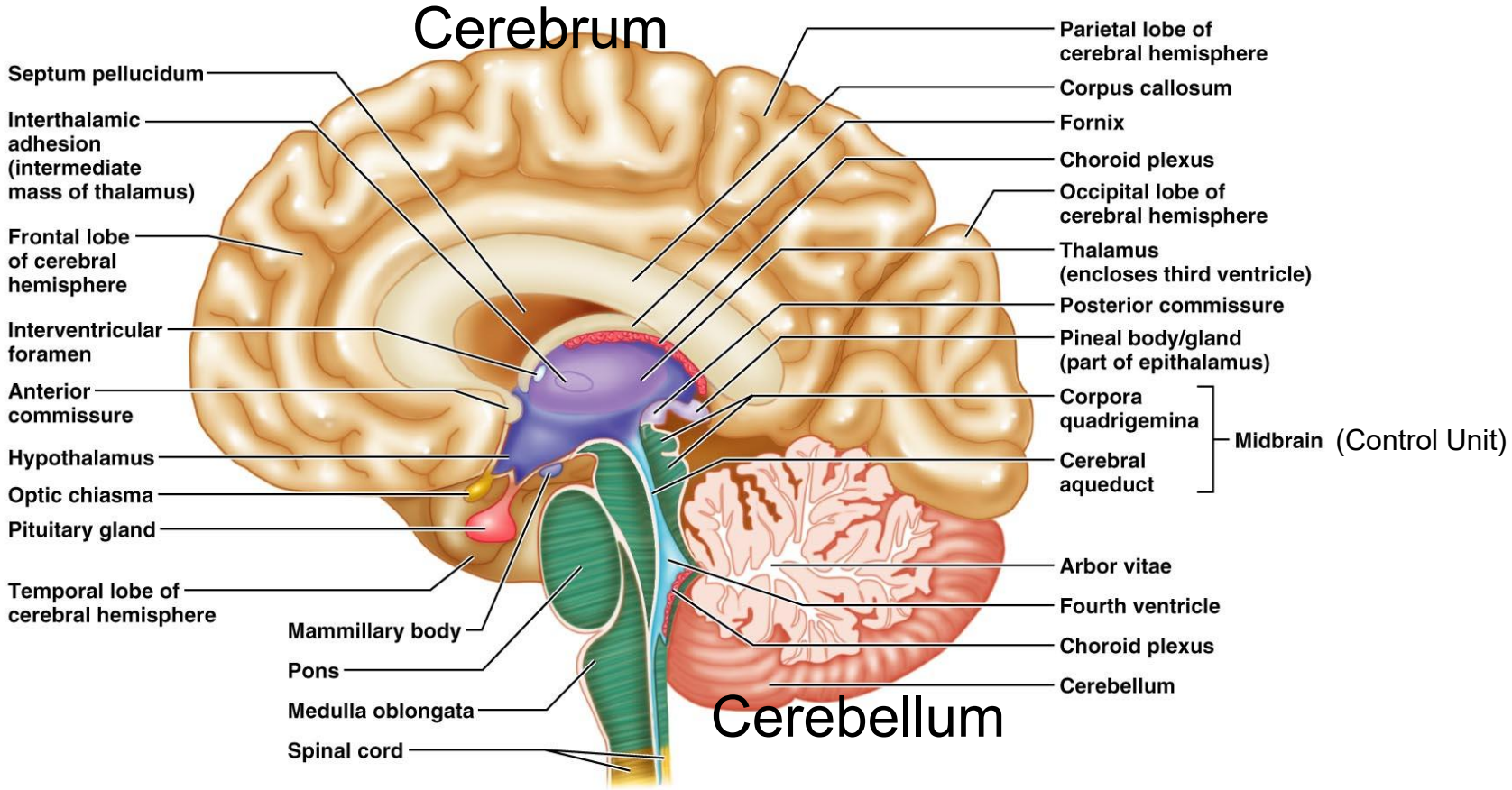


Example

- Person A tells person B: “Please read this question carefully and give me the answer: what is the sum of 6 + 9?”. Explain the mental activities of person B when she or he reads the question and produce the answer.
- Answer:

1. Read and store the question
2. Interpret the meaning of instruction “+”
3. Read the operand 6
4. Read the operand 9
5. Do arithmetic operation of “add”
6. Store the result “15”
7. Synthesize the answer “The sum is 15”
8. Pronounce the answer
9. Wait and listen to the reply from person A

Blueprint of Human Brain



Characteristics of Human Brain+Mind

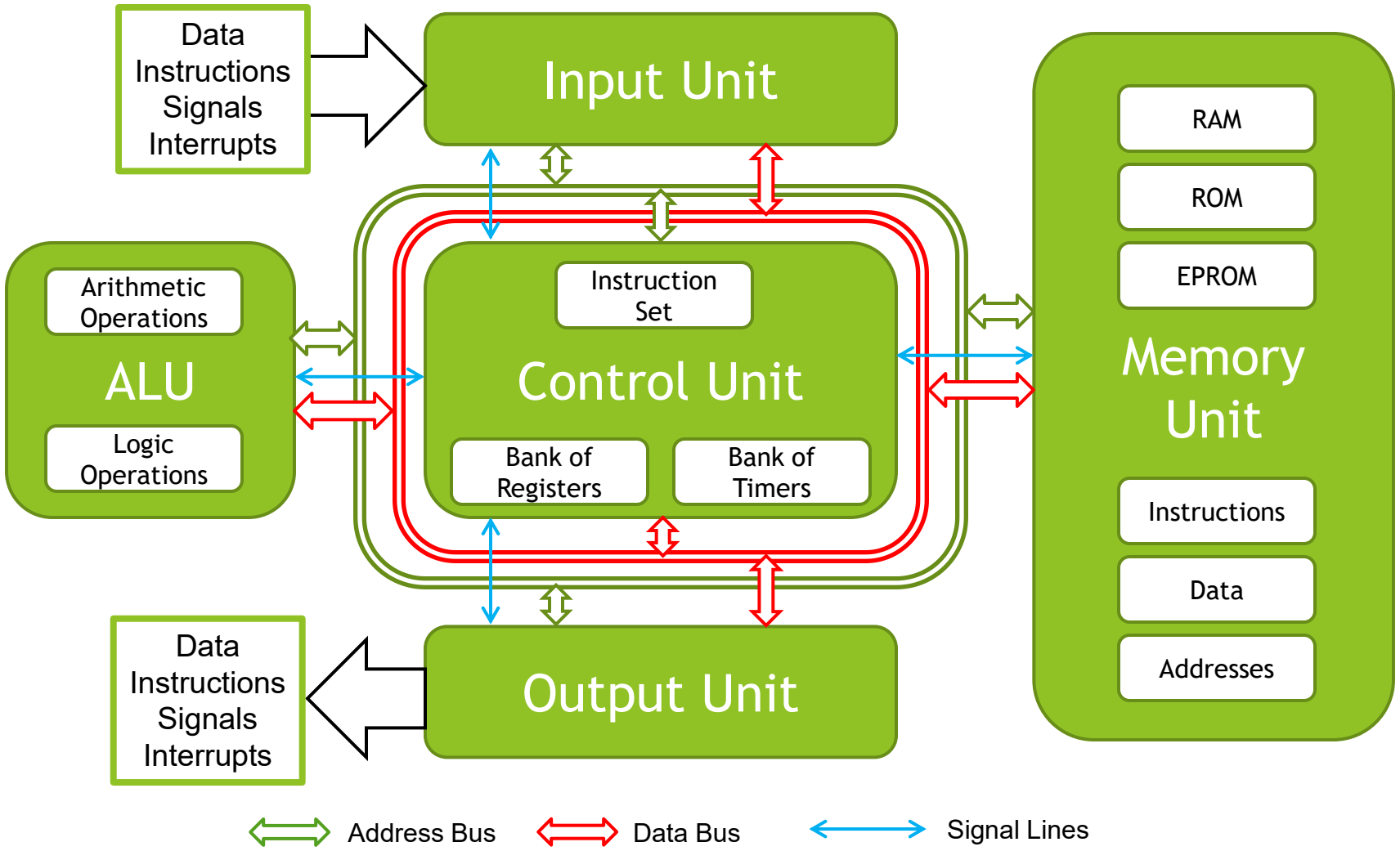
- Is able to memorize past, present and future
- Is able to capture input data
- Is able to undertake operations with numbers
- Is able to undertake operations with numbers/symbols
- Is able to produce output data
- Is able to control mental operations
- Is able to control physical actions
- Is able to communicate
- Is able to learn human languages
- Is able to learn machine languages
- Is able to learn new knowledge and new skills, etc.

Example of Mental Activities

- Person A asks robot B the following question: “Is it lunch time now?”. Explain the mental activities of robot B when it prepares the reply.
- Answer:

1. Read the current time T
2. Compare it with 12h00
3. If T > 12h00, compare it with 14h00
4. If 14h00 > T > 12h00, reply “yes”
5. Otherwise, reply “not yet”

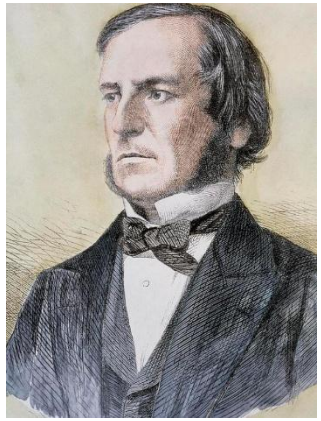
Blueprint of Machine's Brain



Characteristics of Machine's Brain+Mind

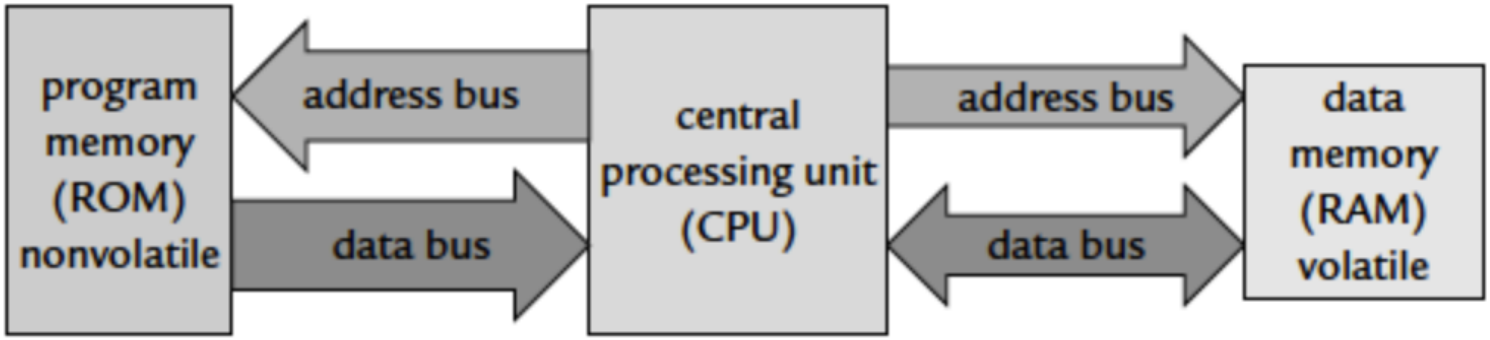
- Is able to memorize data (to use) and instructions (to do)
- Is able to capture input data
- Is able to undertake arithmetic operations with numbers
- Is able to undertake logic operations with numbers/symbols
- Is able to produce output data
- Is able to control mental operations
- Is able to control physical actions
- Is able to communicate
- Is able to learn human languages (future)
- Is able to learn machine languages (future)
- Is able to learn new knowledge and new skills, etc. (future)

Blueprints of Machine's Brains

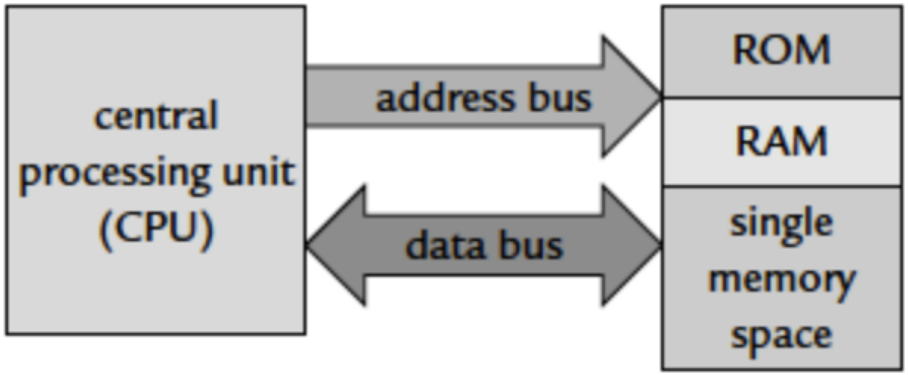


George Boole
1815 - 1864

(a) Harvard architecture



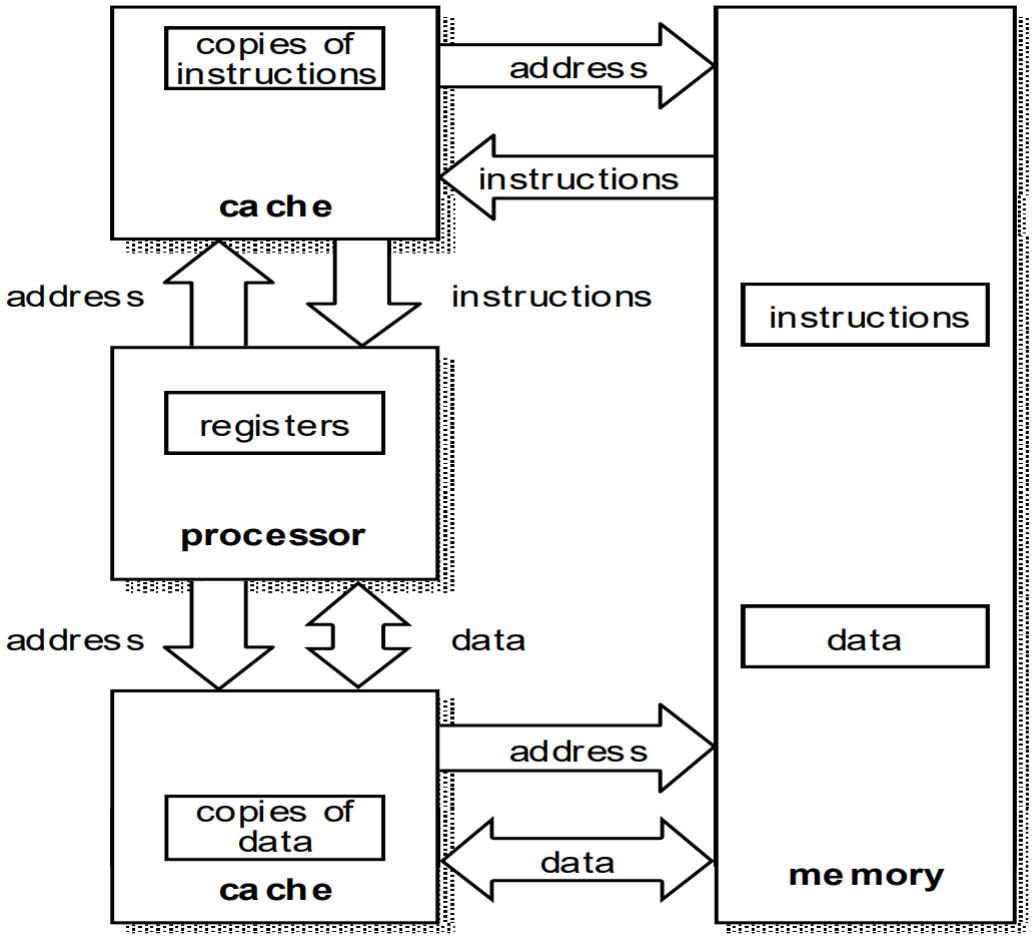
(b) von Neumann architecture

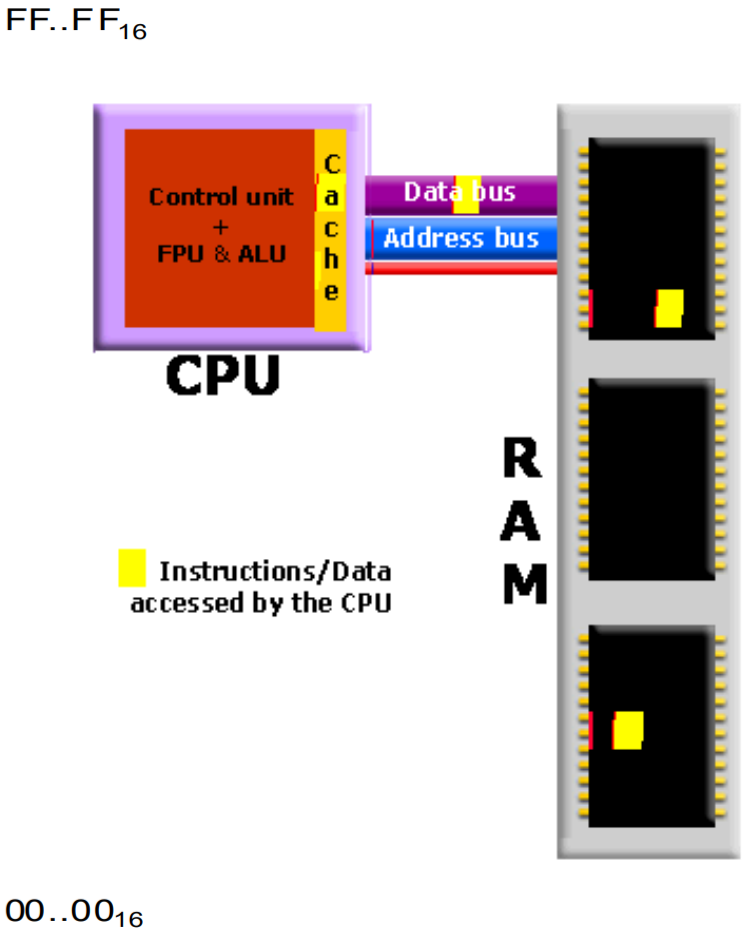


John von Neumann
1903 - 1957

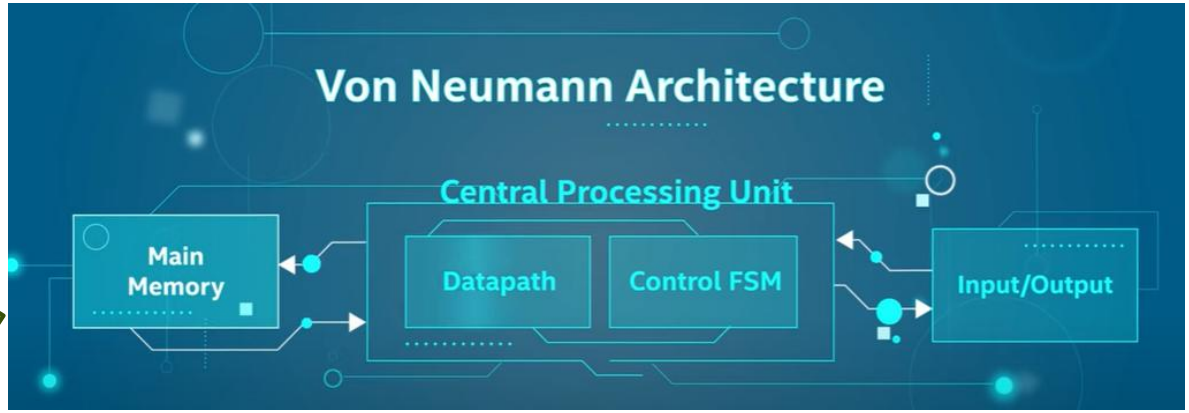
Alan Turing
1912 - 1954

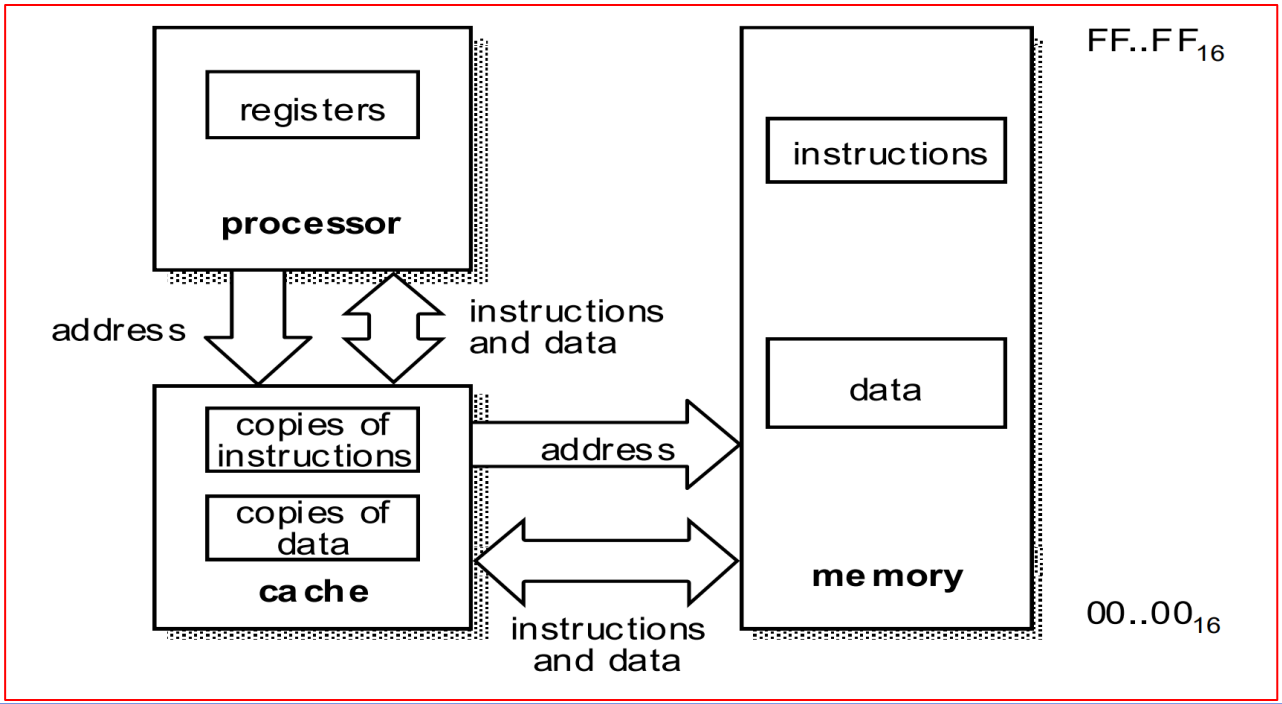
Blueprint of Harvard Architecture



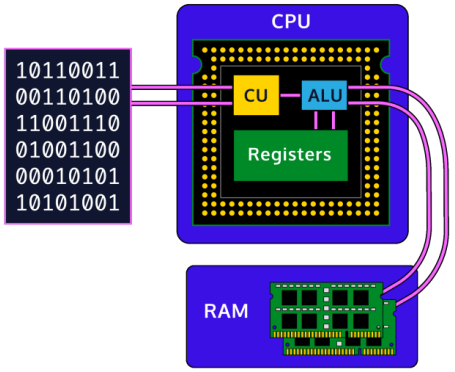


Blueprint of John von Neumann Architecture

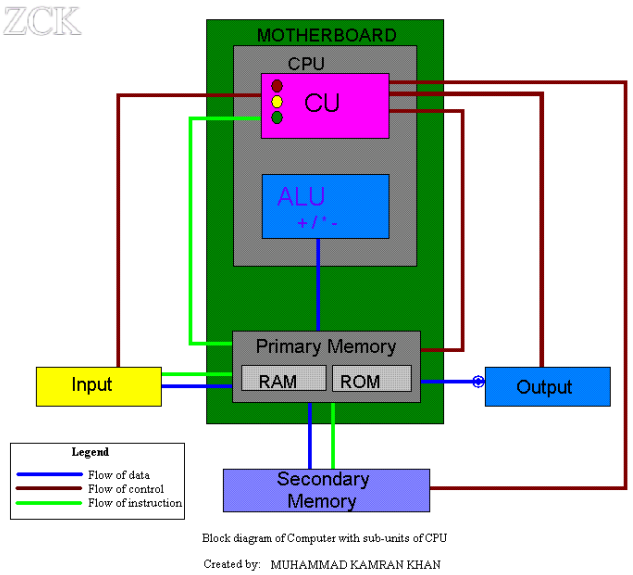




Outline



- Basics of Brains
- Inventors of Digital Computers
- Family of ARM Microprocessors
- Integrated Development Environment



Who is the father of computer?

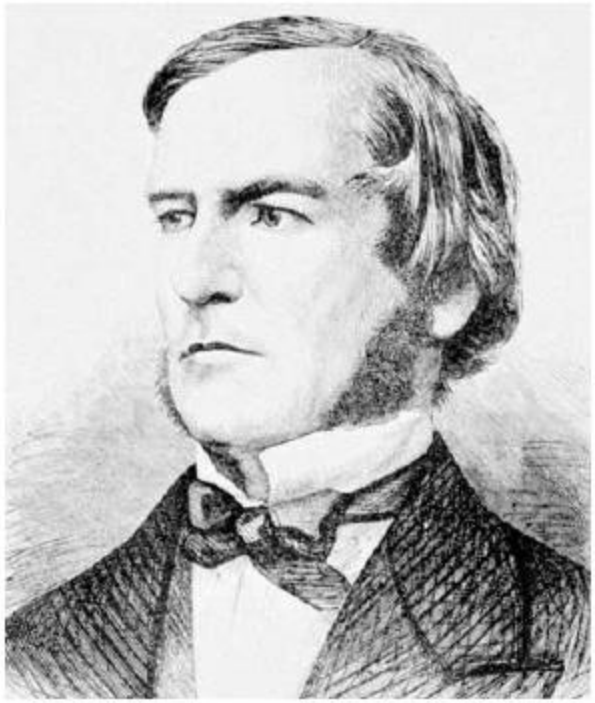
- **The father of computer is Charles Babbage.**

English inventor
1791–1871
taught math at Cambridge University
invented a viable mechanical computer equivalent to modern digital computers

02:00 22/04/2015

George Boole

- Born 2 November 1815
- Died 8 December 1864
- Born in Lincoln, Lincolnshire, England
- A mathematician and philosopher, who invented Boolean logic – the basis of modern digital computer logic



John von Neumann

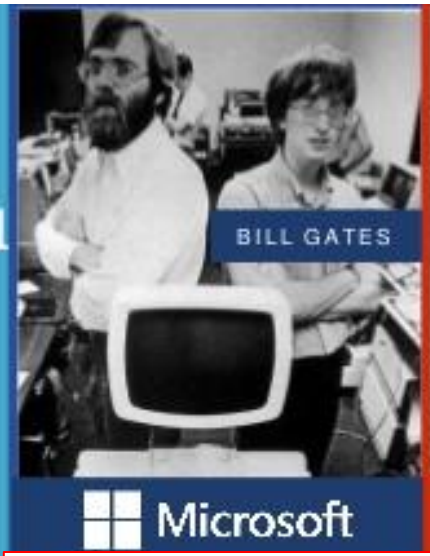
- Born December 28, 1903
- Died February 8, 1957
- Born in Budapest, Austria-Hungary
- A mathematician who made major contributions to set theory, functional analysis, quantum mechanics, ergodic theory, continuous geometry, economics and game theory, computer science, numerical analysis, hydrodynamics and statistics.

- 1968- ROBERT NOYCE- one of the inventors of integrated circuits founded the INTEL.
- 1969- ARPANET is set up later becomes INTERNET.
- 1972-THE C programming language is developed at AT&T Bell Labs by Brian Kerningham and Dennis Ritchie.
- UNIX is written at Bell Labs and become the most portable operating system.

MICROSOFT & APPLE 1976

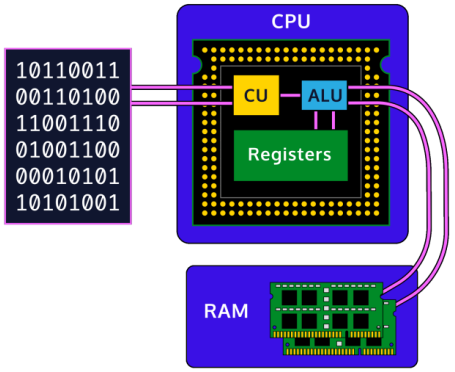
INVENT THE PERSONAL COMPUTER (PC)

- In 1975 they were the first computers in homes
- Windows computer invented by Bill Gates (Microsoft)
- Apple computer by Steve Jobs (Apple)

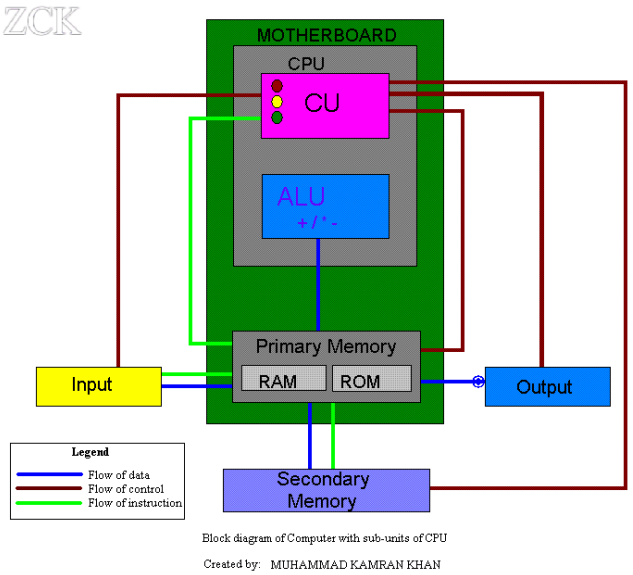


When will we see the emergence of Personal Robot?

Outline



- Basics of Brains
- Inventors of Digital Computers
- Family of ARM Microprocessors
- Integrated Development Environment



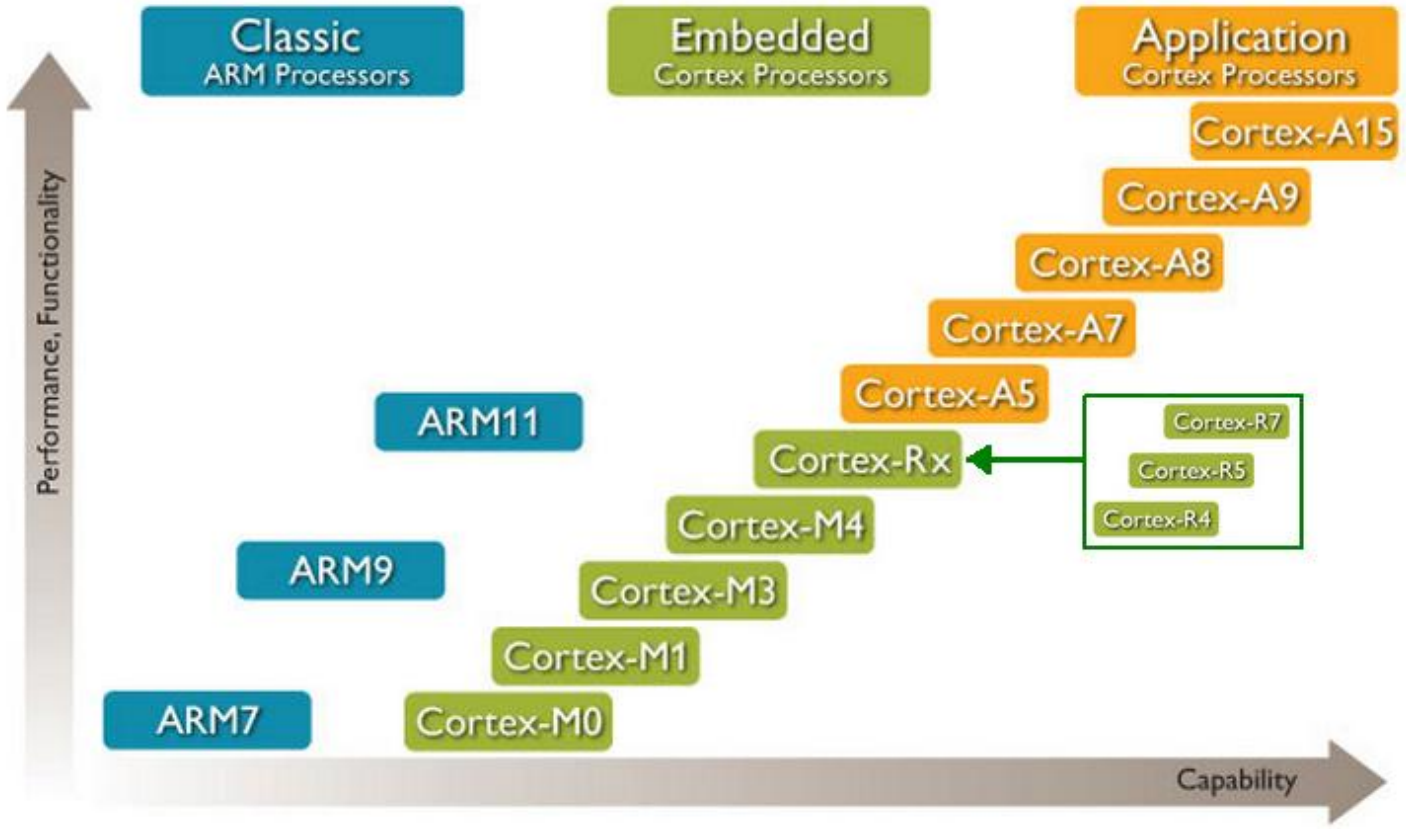
(RISC=Reduced Instruction Set Computer)

ARM: Advanced RISC Machines

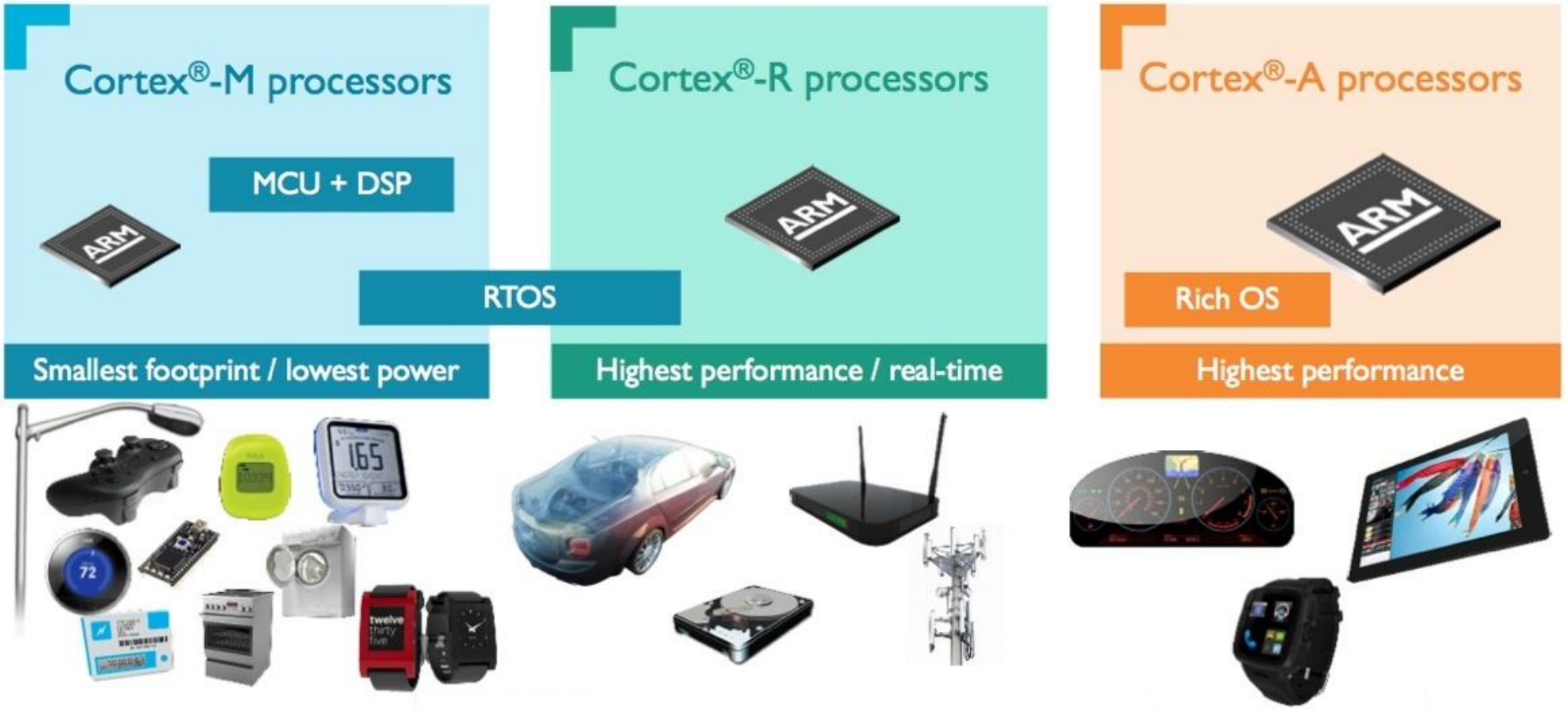
- ARM Ltd was founded in 1990
- ARM Ltd is headquartered in Cambridge, UK

ARM Family

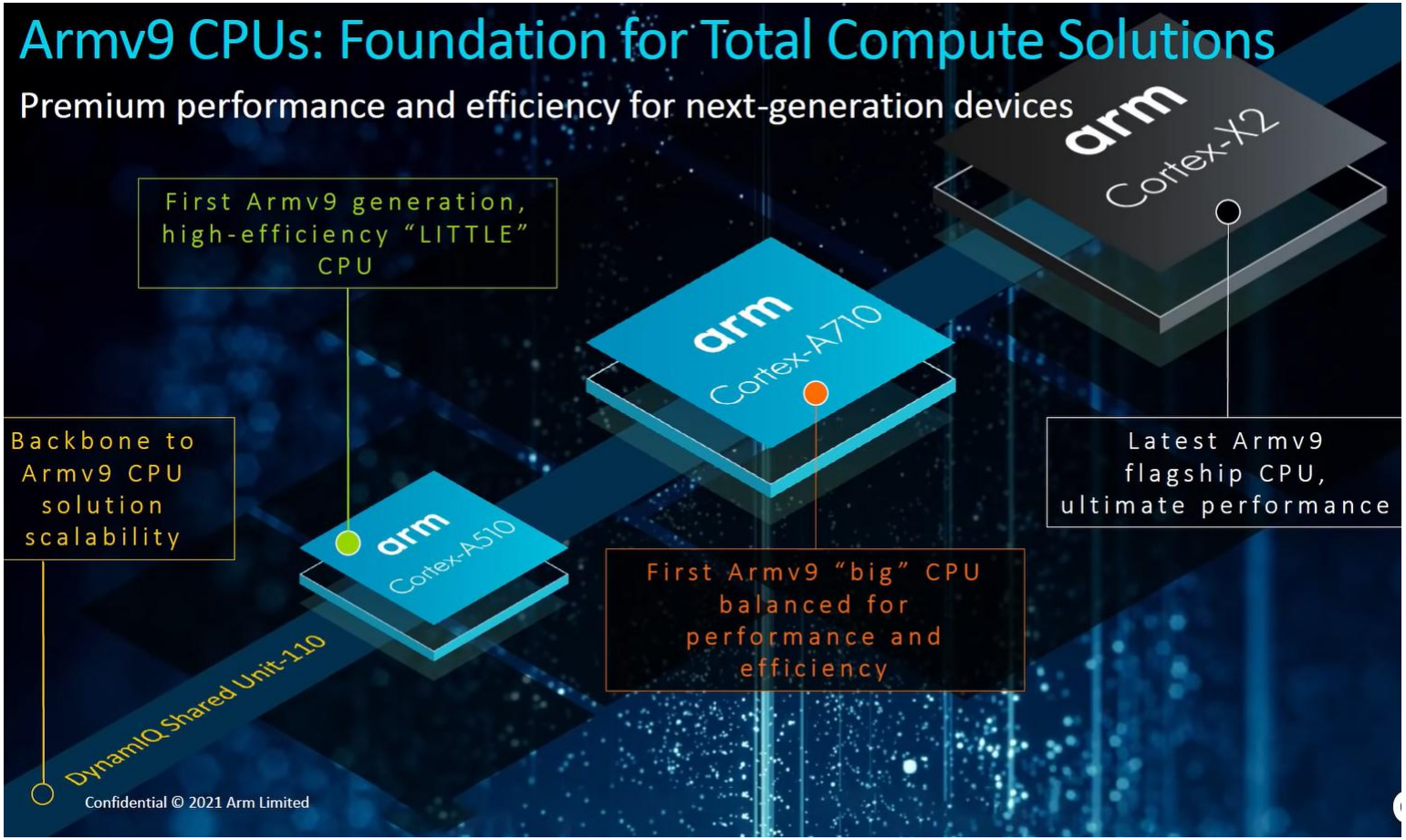
- Cortex-M Series: for microcontroller applications
- Cortex-R Series: for real-time applications
- Cortex-A Series: for advanced applications



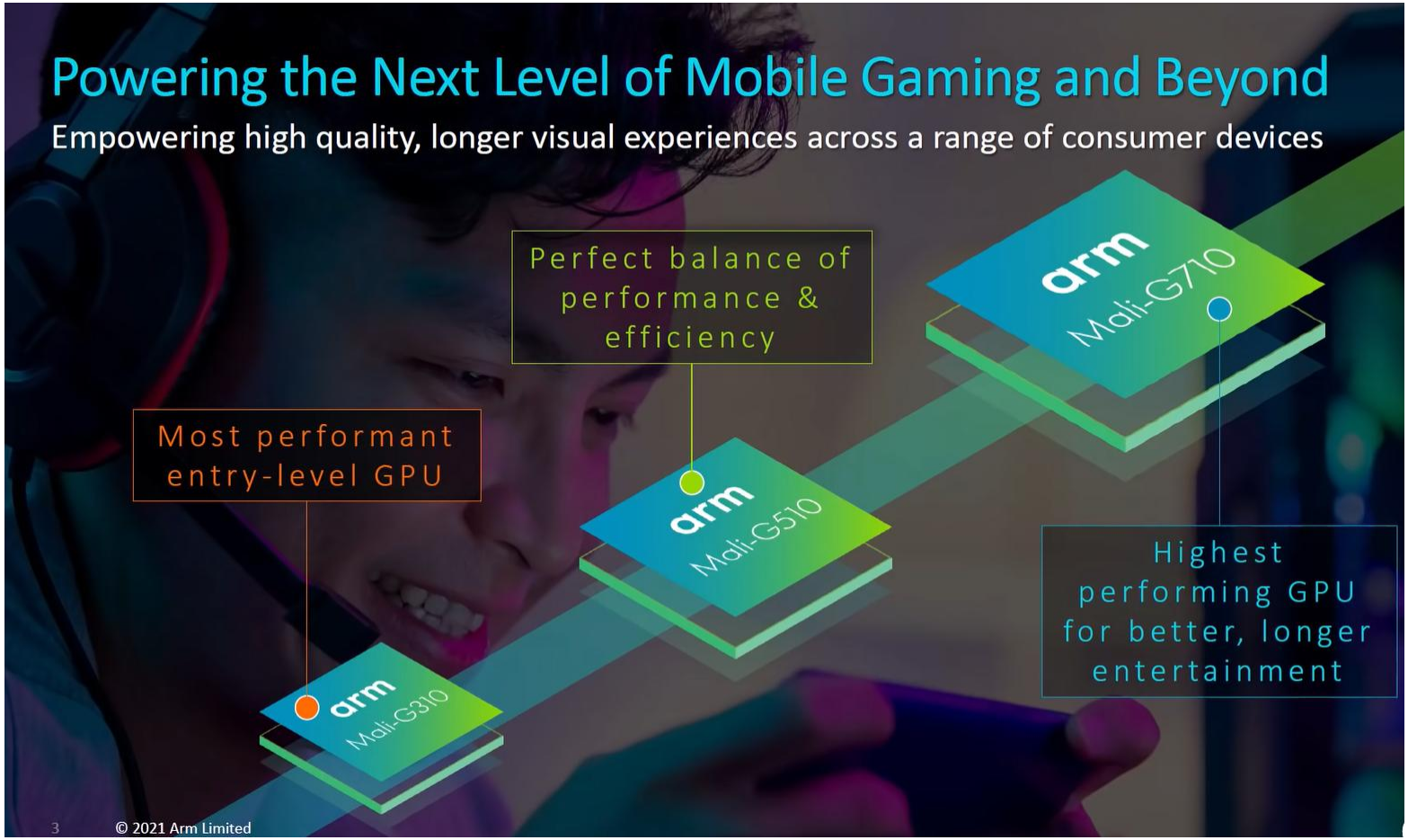
Example of Applications …



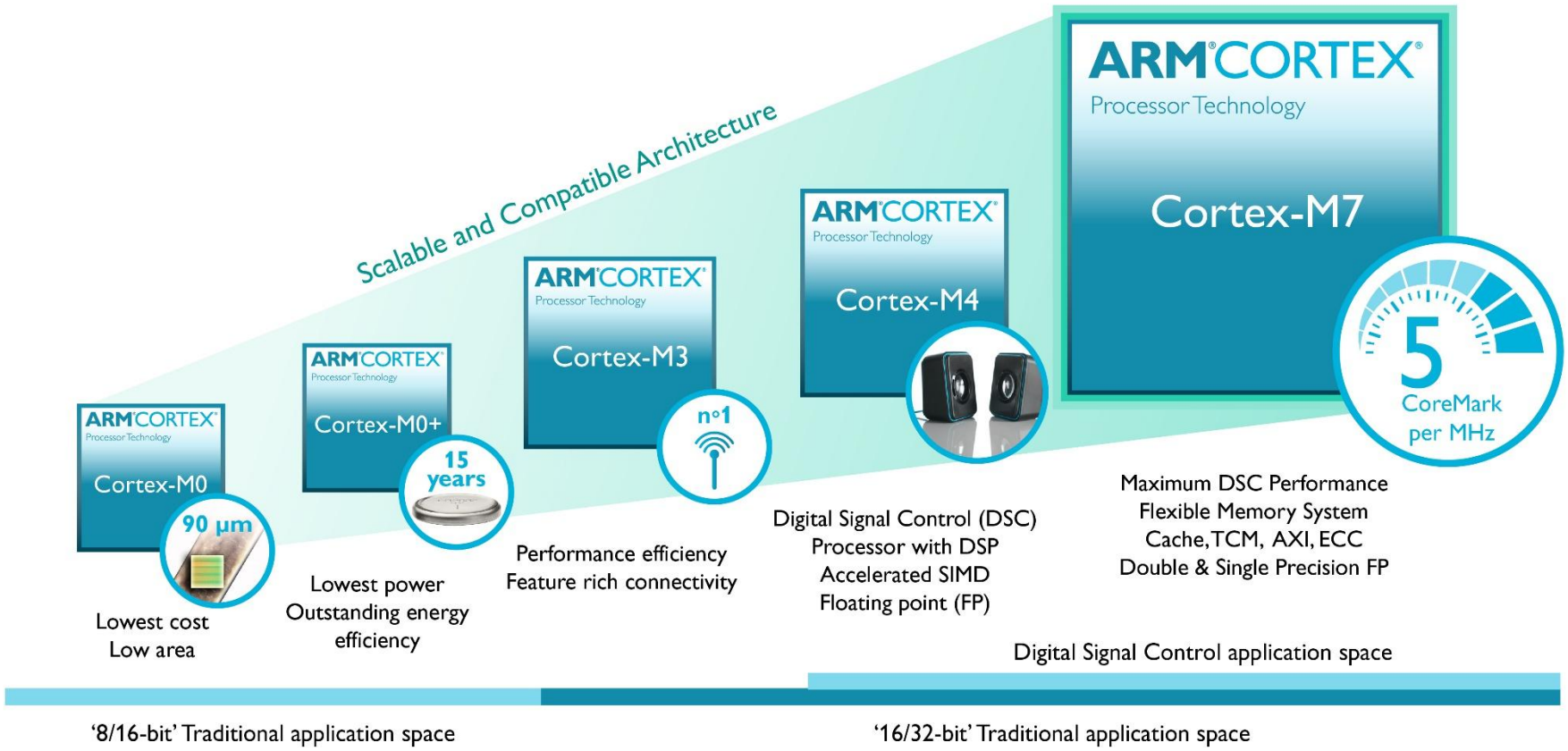
Evolution of ARM Family



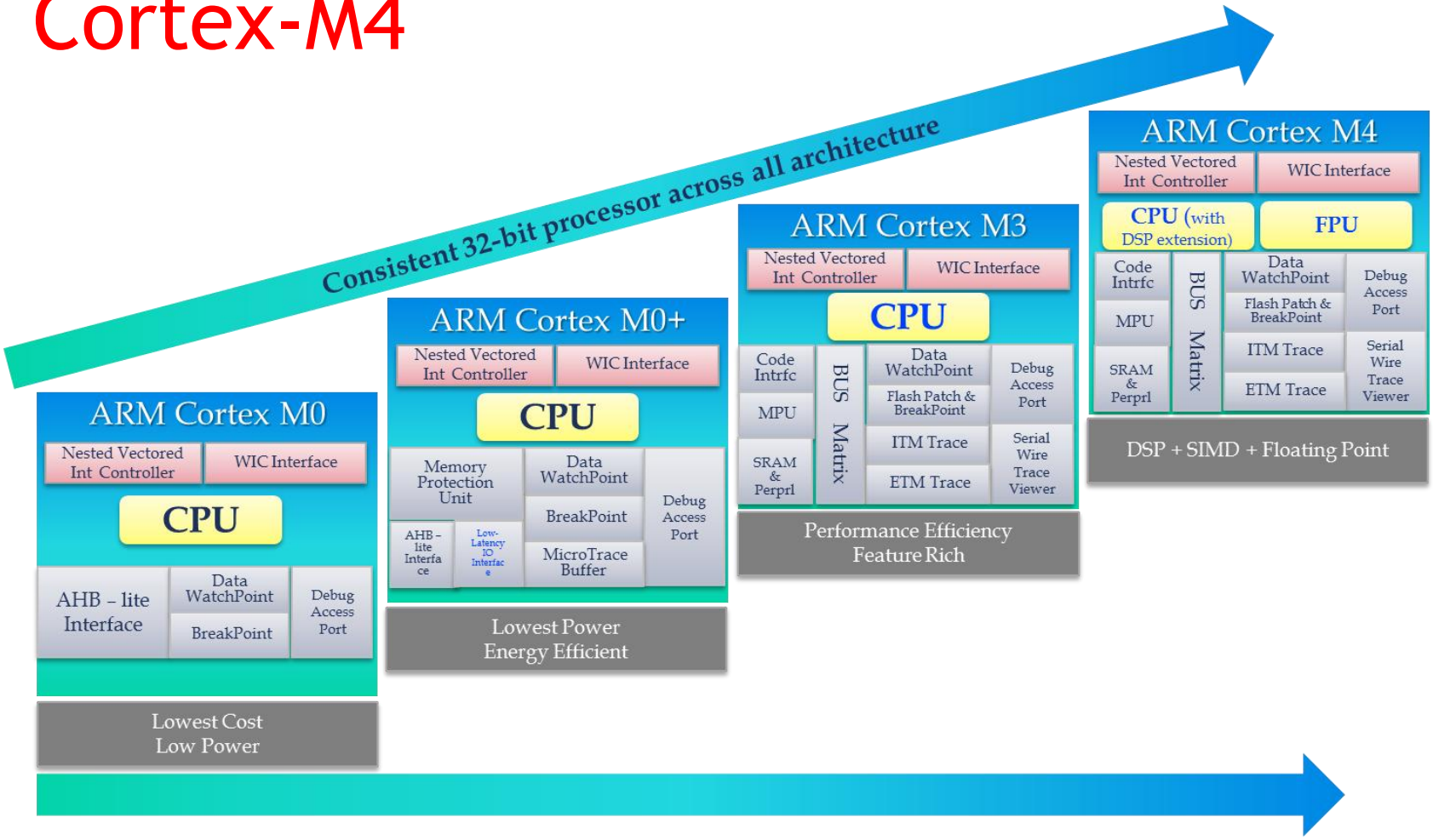
Evolution of ARM Family (continued)



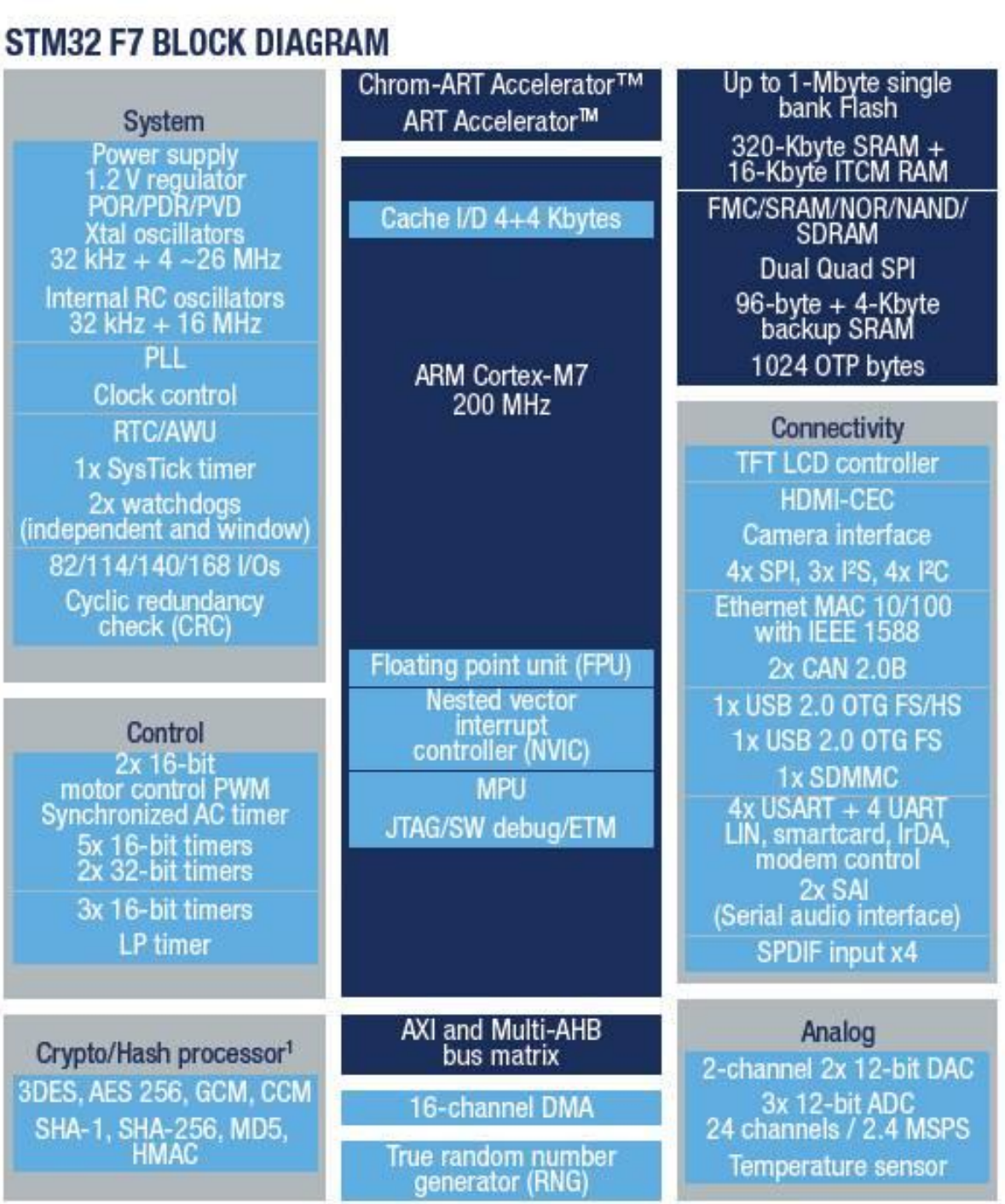
Evolution of ARM Cortex-M Series



Example: Cortex-M4



Example: Cortex-M7



ARM's Worldwide Partners

ARM's Development Boards

Example: Tiva C Series TM4C123G

TM4C123x

Temperatures	85°C	105°C

ARM® Cortex®-M4 Up to 80 MHz

FPU	MPU
NVIC	ETM	SWD/T

Memory
- Up to 256 KB Flash
- Up to 32 KB SRAM
- 2 KB EEPROM
- ROM
- DMA (32 ch)

Power & Clocking
- Precision Oscillator
- RTC Battery-Backed Hibernate

System Modules
- 6× 32-bit Timer/PWM/CCP
- 6× 64-bit Timer/PWM/CCP
- Systick Timer
- 2× Watchdog Timer

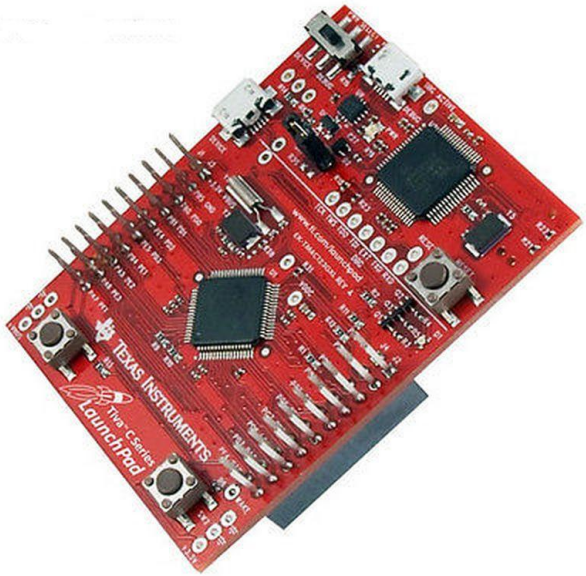
Debug
- Real-time JTAG

Control Peripherals
- 2× Quadrature Encoder Inputs
- 16× PWM Outputs

Comms Peripherals
- 8× UART
- 4× SSI/SPI
- 6× I²C
- 2× CAN
- USB Full Speed (Host/Device/OTG)

Analog
- 2× 12ch, 12-bit ADCs, 1MSPS
- LDO Voltage Regulator
- 3× Analog Comparators
- Temperature Sensor

FPU: Floating-point Unit
MPU: Multi-core Processing Unit



3 Choose Your LaunchPad | 2 Download Software | 1 Plug in BoosterPack

MSP LaunchPads

C2000 LaunchPads

Connected LaunchPads

Hercules LaunchPads

EK-TM4C123GXL

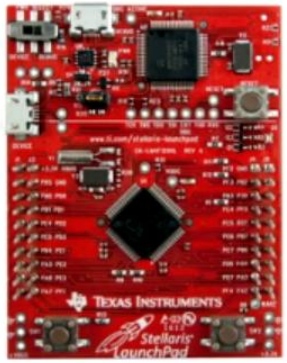
Project Zero: Your First Project

Introduction

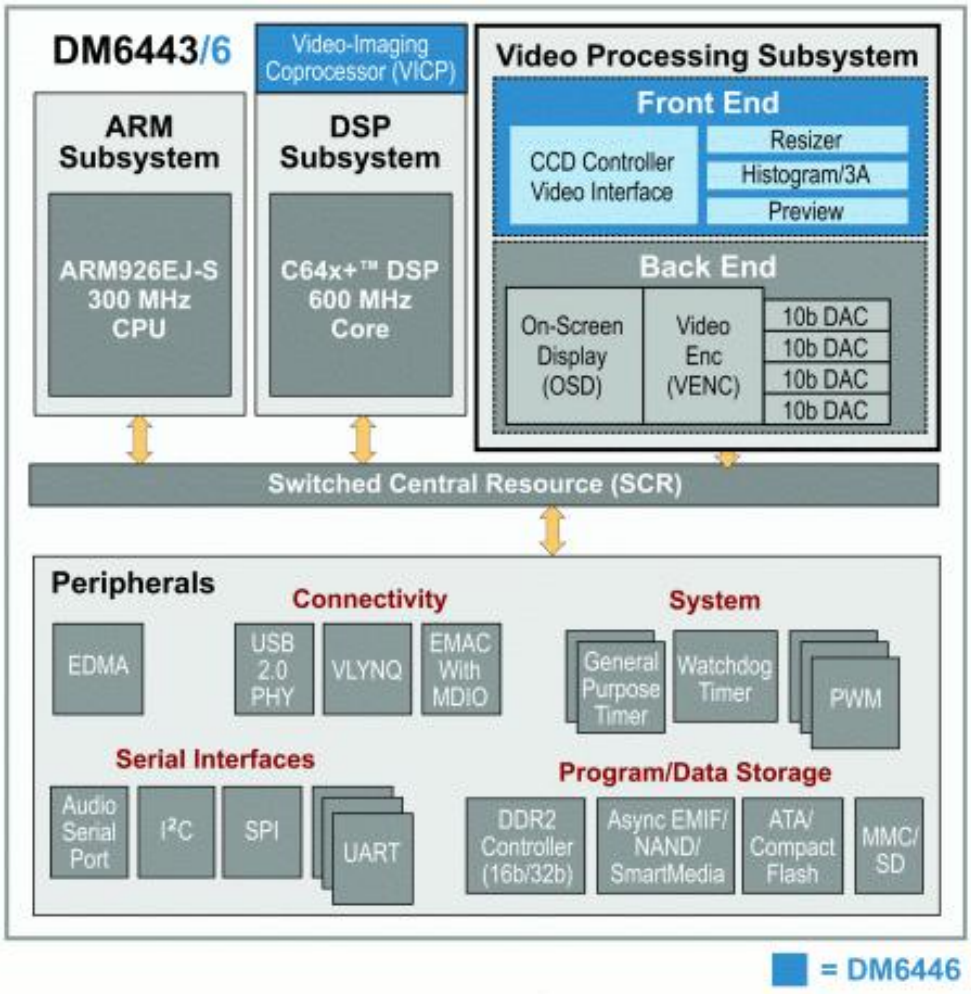
Note that the video below uses the Stellaris LM4F120 LaunchPad, but the content of the video is equally applicable to the Tiva TM4C123G LaunchPad.

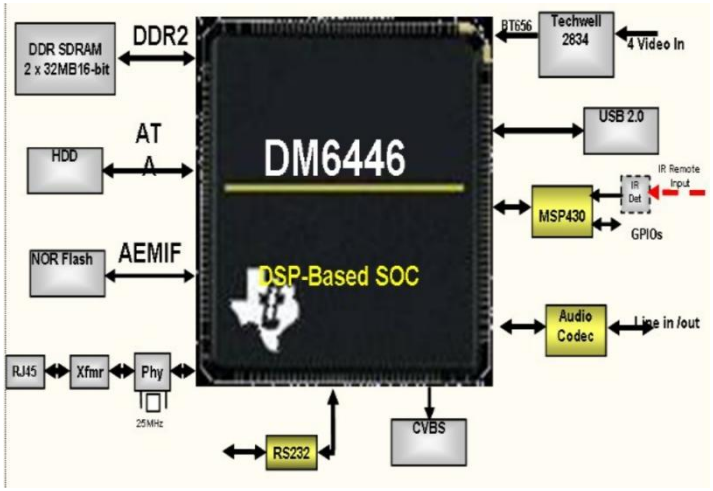
Stellaris® LaunchPad

- ARM® Cortex™-M4F 64-pin 80MHz LM4F120H5QR
- On-board USB ICDI (In-Circuit Debug Interface)
- Micro AB USB Device port
- Device/ICDI power switch
- BoosterPack XL pinout also supports existing BoosterPacks
- 2 user pushbuttons
- Reset button
- 3 user LEDs (1 tri-color device)
- Current measurement test points
- 16MHz Main Oscillator crystal
- 32kHz Real Time Clock crystal
- 3.3V regulator
- Support for multiple IDEs:

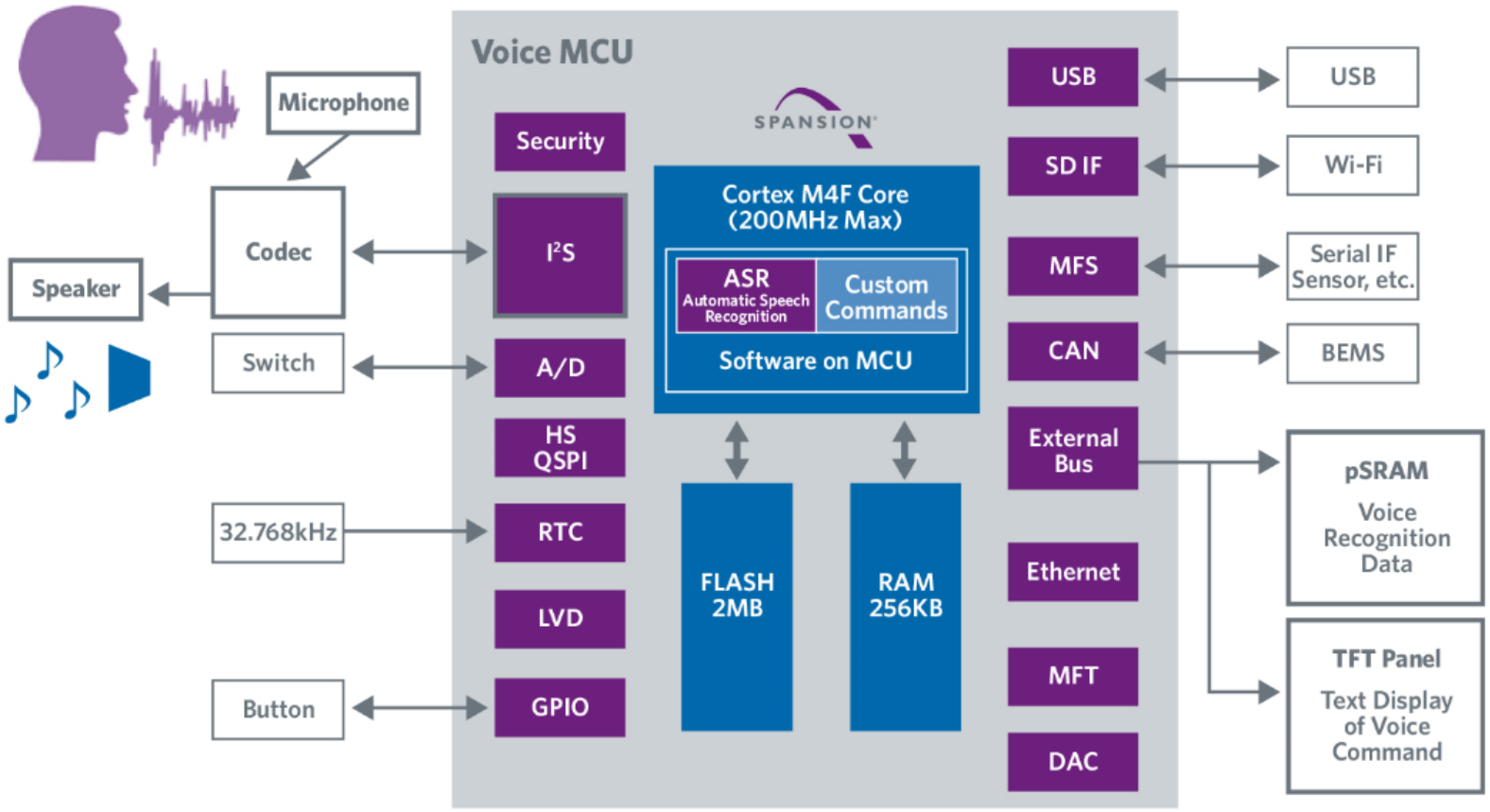


Example of ARM for Vision





Example of ARM for Speech

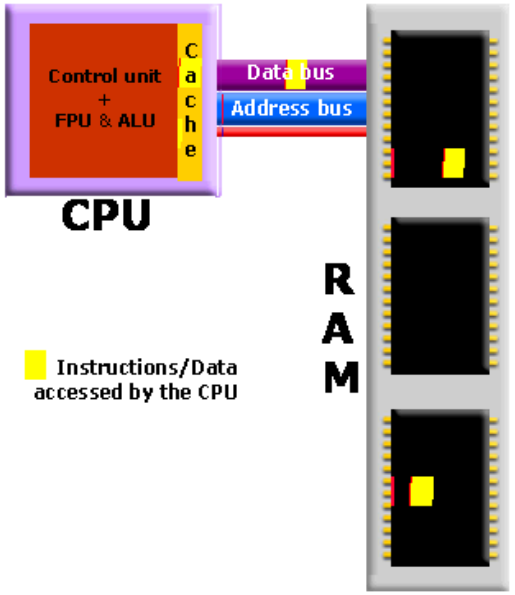


ARM's Software Development Tools

- Keil www.keil.com
- Energia www.energia.nu
- Code Composer Studio IDE www.ti.com/ccs

ARM Cortex is RISC architecture

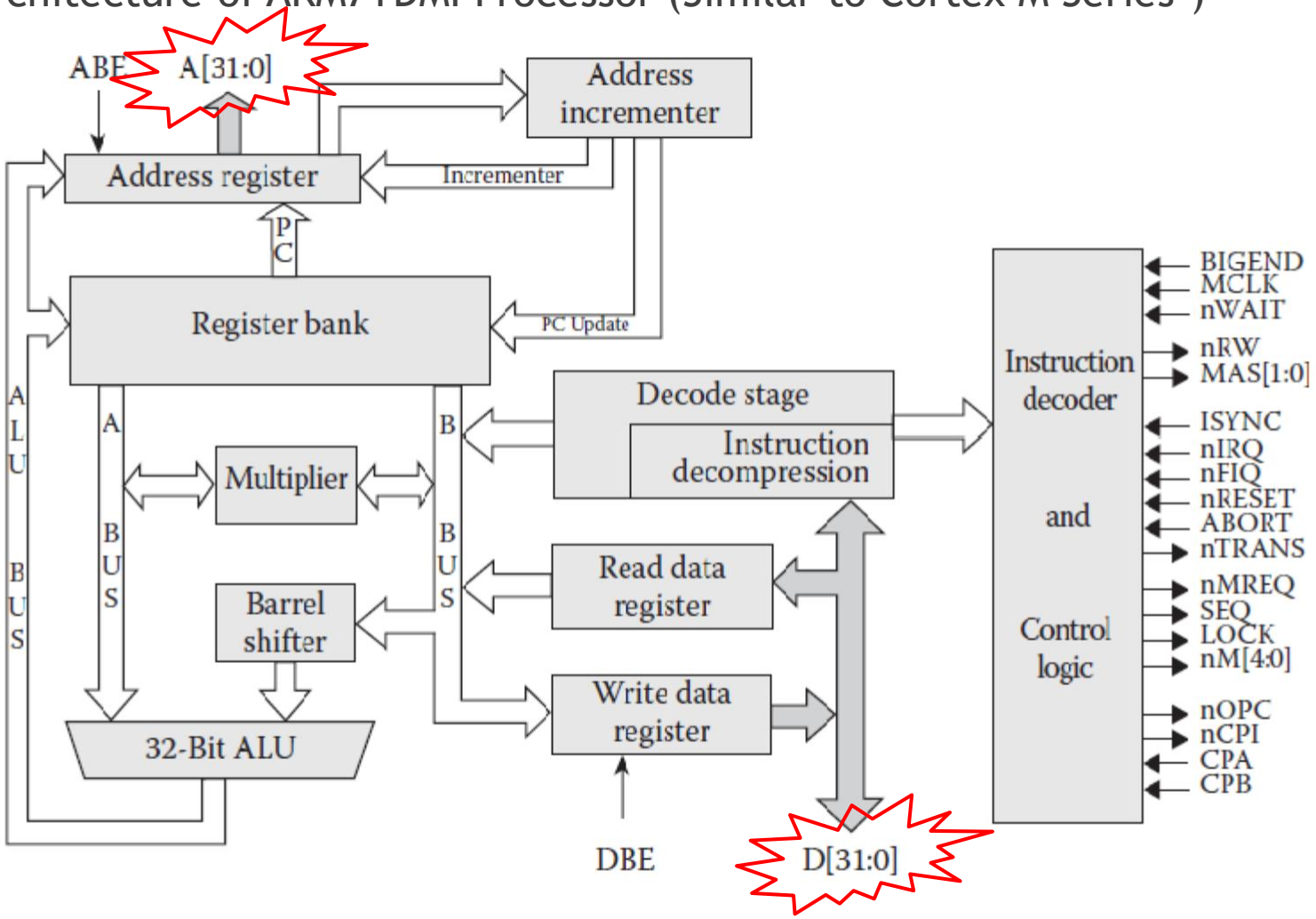
- ARM is a RISC architecture (RISC=Reduced Instruction Set Computer)
  - ISA stands for Instruction Set Architecture which are innate
  - RISC stands for Reduced Instruction Set Computer
  - Most instructions execute in a single cycle
  - ARM Instruction Set is 32-bit
  - Thumb Instruction Set is 16/32-bit



- ARM is a 32-bit load-store architecture
  - 8 bits are called a Byte
  - 16 bits or two bytes are called a Halfword
  - 32 bits or four bytes are called a Word
  - 64 bits or eight bytes are called Double-word
  - The only memory accesses are loads and stores

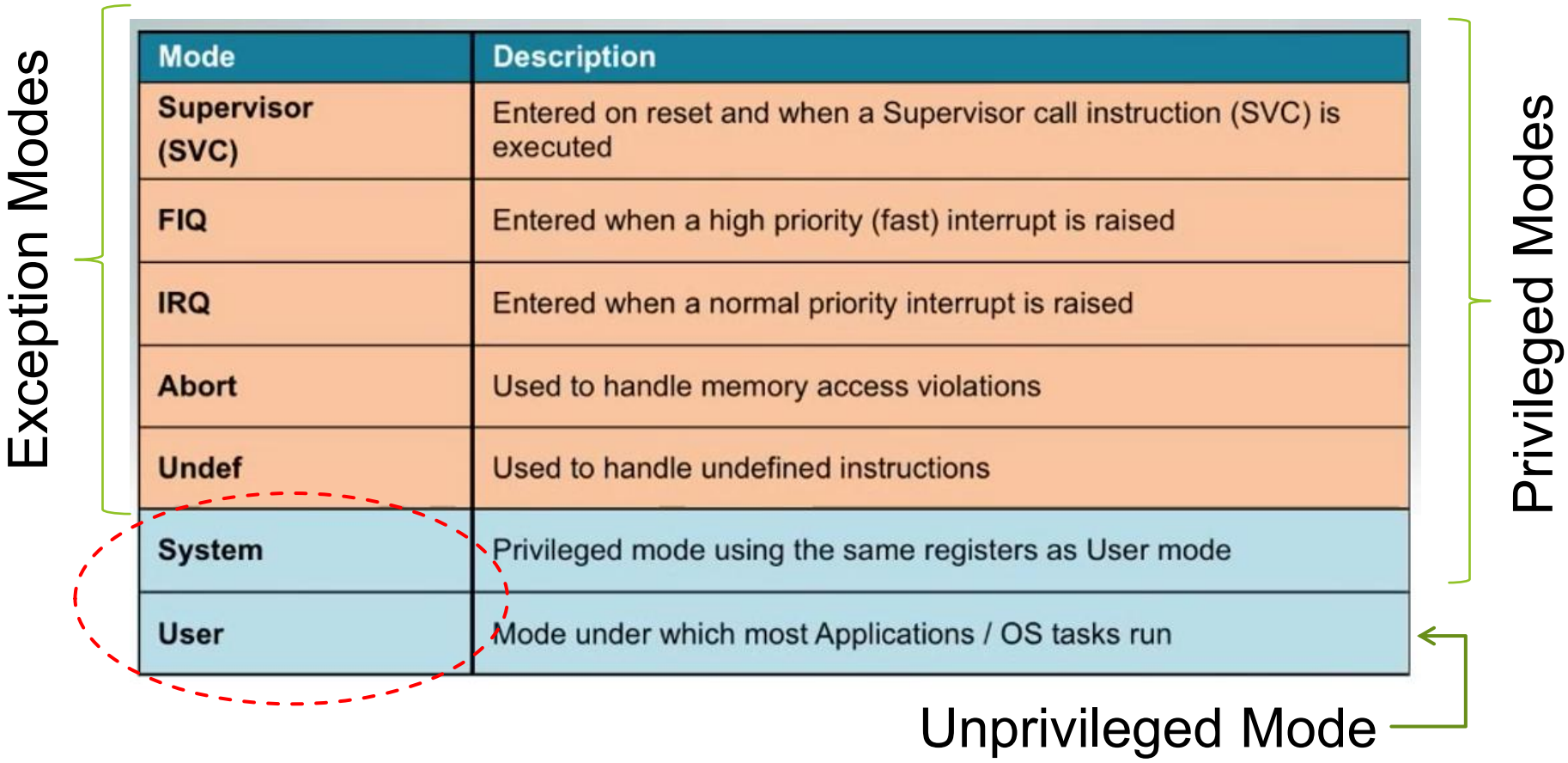
ARM Cortex is 32-bit microcontroller

- Architecture of ARM7TDMI Processor (Similar to Cortex M Series')



ARM Cortex supports System mode and User mode

- ARM Core has seven basic modes

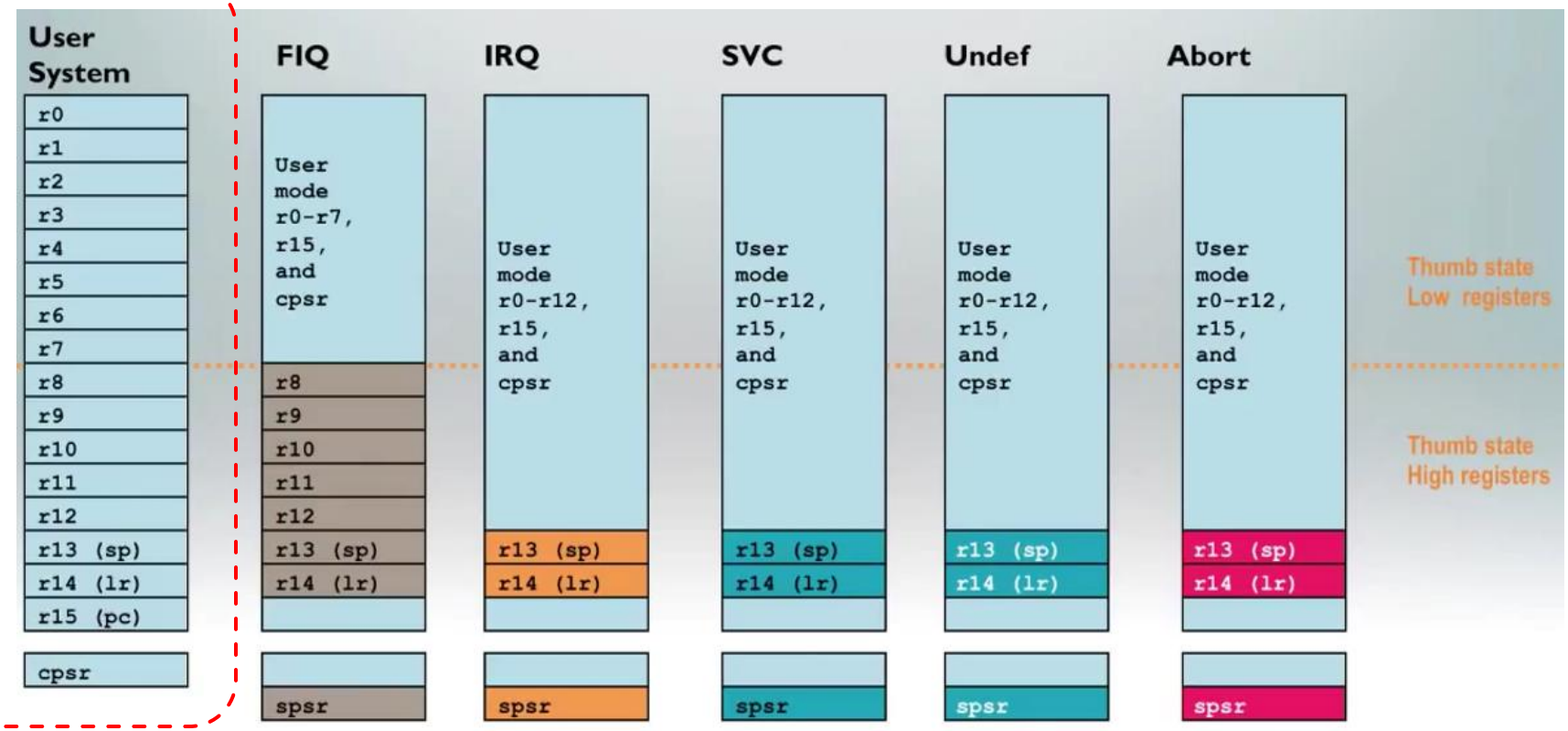


Mode	Description
Supervisor (SVC)	Entered on reset and when a Supervisor call instruction (SVC) is executed
FIQ	Entered when a high priority (fast) interrupt is raised
IRQ	Entered when a normal priority interrupt is raised
Abort	Used to handle memory access violations
Undef	Used to handle undefined instructions
System	Privileged mode using the same registers as User mode
User	Mode under which most Applications / OS tasks run

ARM Cortex has over 16 registers

- ARM Core has a large number of registers

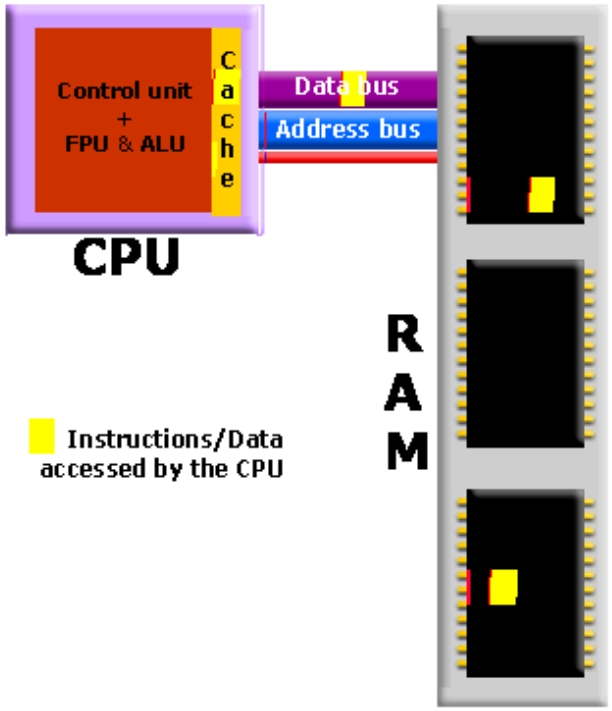
SPSR: Saved Program Status Register
CPSR: Current Program Status Register



Note: System mode uses the User mode register set

SP: Stack Pointer
LR: Link Register

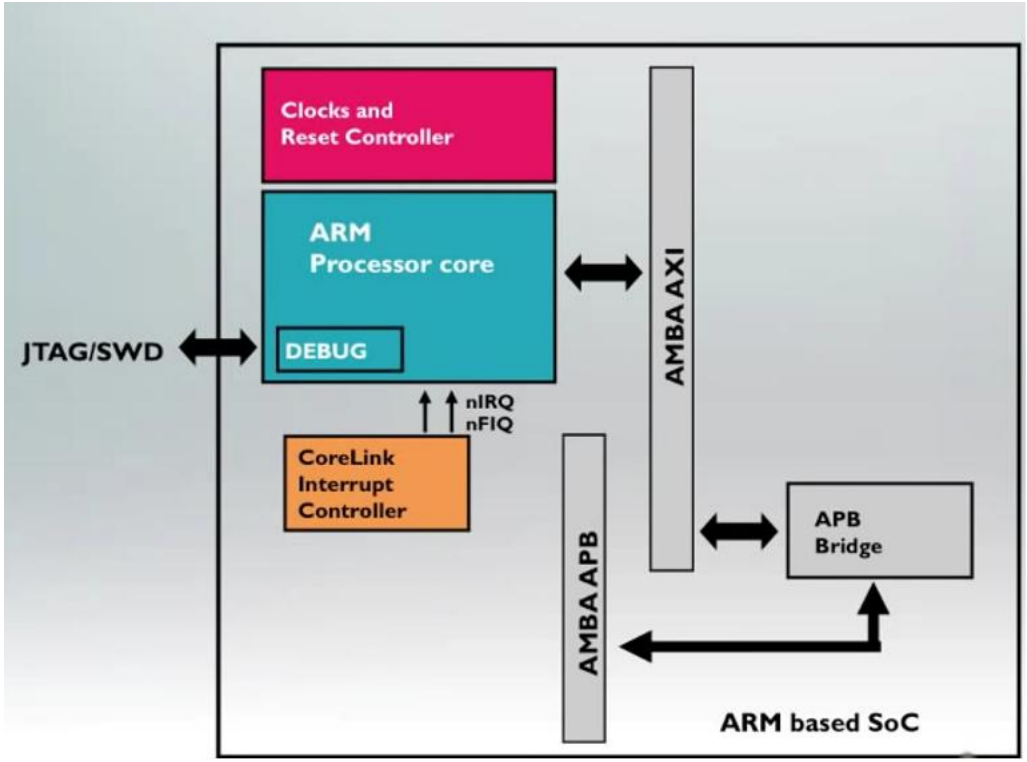
ARM Cortex could serve as microcontroller



- ARM supports Application Profile
  - Memory Manage Unit (MMU)
  - Highest performance at lowest power consumption
  - Trust zone for a safe and extensible system
- ARM supports Real-time Profile
  - Memory Protection Unit (MPU)
  - Low latency and predictability of real-time needs
  - Tightly coupled memory for fast and deterministic access
- ARM supports Microcontroller Profile
  - Lowest gate count entry point
  - Deterministic and predictable behaviour
  - Deeply embedded use

ARM has two internal buses

- The ARM **Advanced Microcontroller Bus Architecture** (**AMBA**) is an open-standard, on-chip interconnect specification for the connection and management of functional blocks in system-on-a-chip (SoC) designs. It facilitates development of multi-processor designs with large numbers of controllers and peripherals.

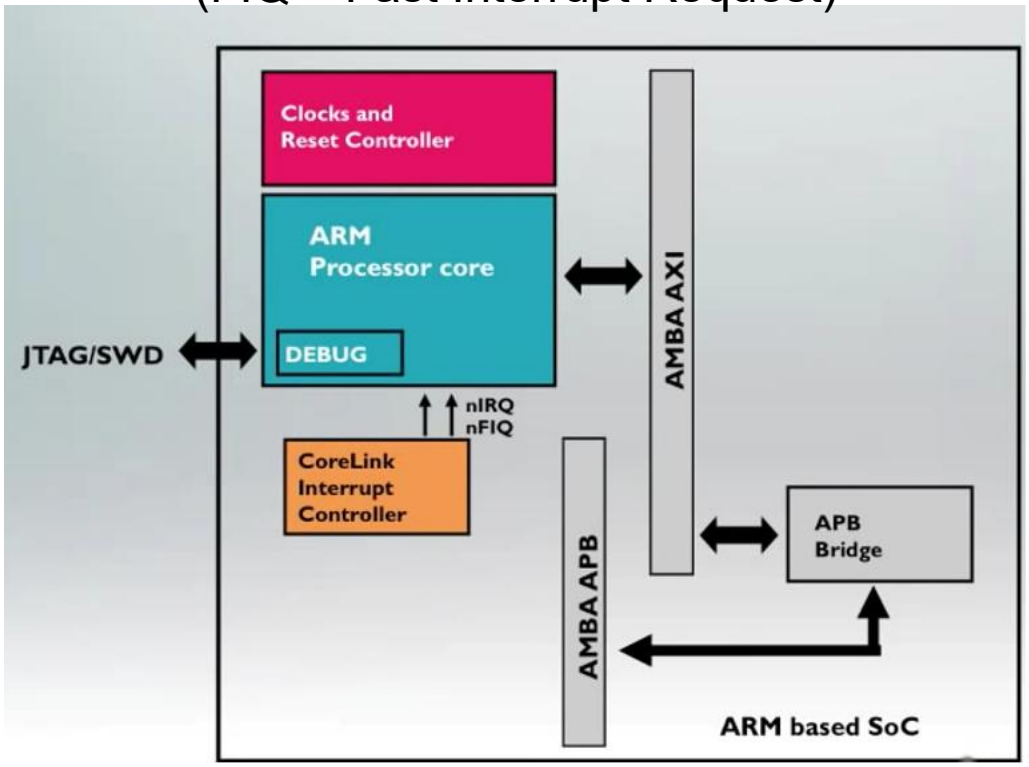


APB = Advanced Peripheral Bus AHB = Advanced High-performance Bus

ARM Cortex supports Fast Interrupt Request

- An **FIQ** is just a higher priority interrupt request, that is prioritized by disabling IRQ and other **FIQ** handlers during request servicing. Therefore, no other interrupts can occur during the processing of the active **FIQ** interrupt.

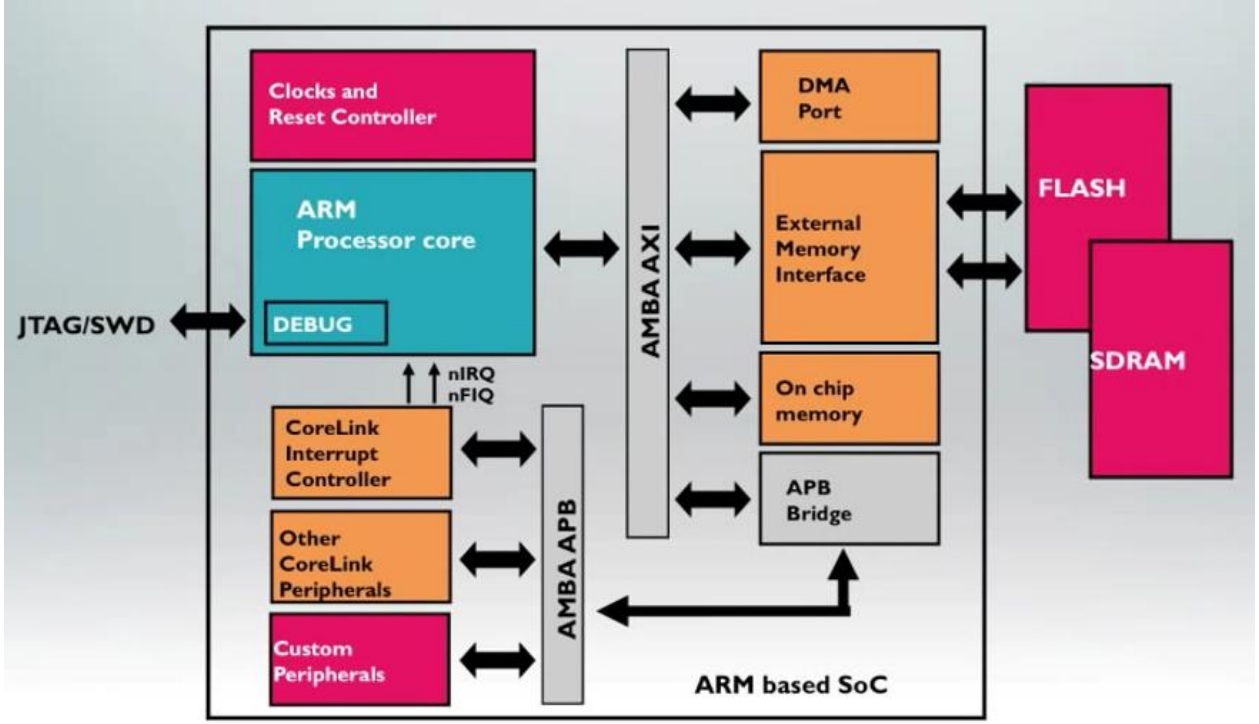
(FIQ = Fast Interrupt Request)



APB = Advanced Peripheral Bus AHB = Advanced High-performance Bus

ARM Cortex supports Direct Memory Access

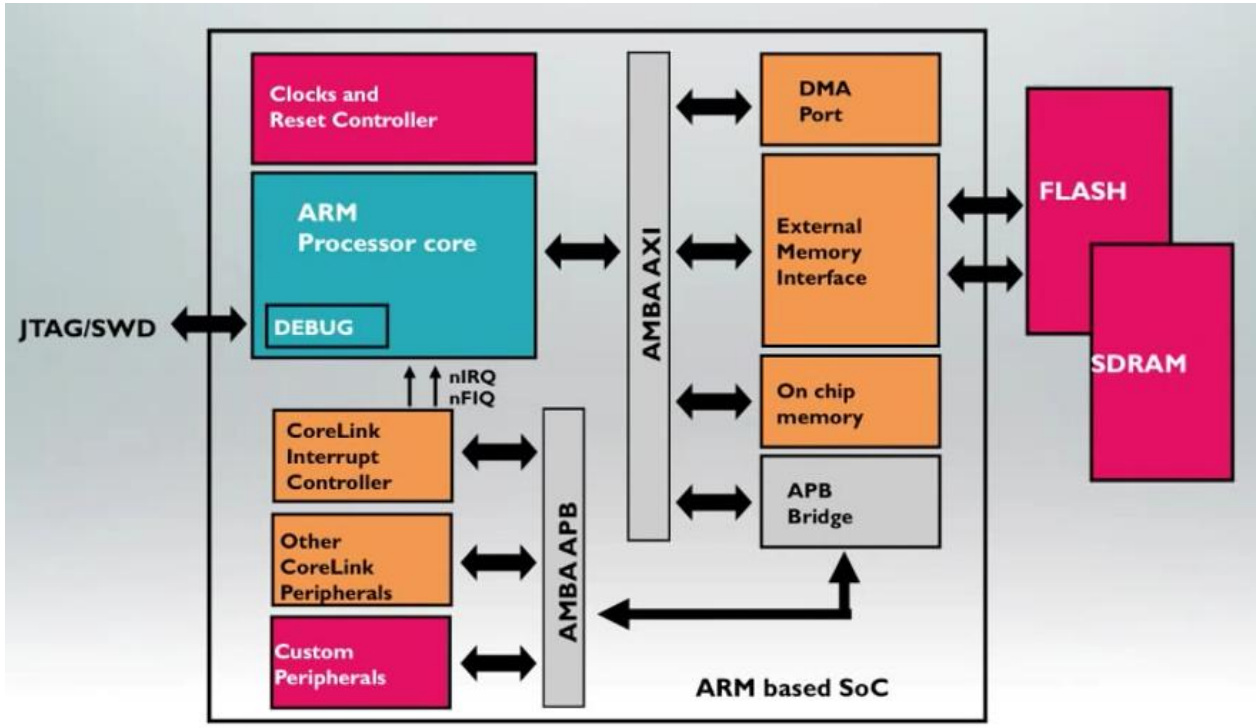
- Direct memory access (**DMA**) is a feature of microprocessor systems that allows certain hardware subsystems to access main system memory such as RAM (Random Access Memory) independently of the central processing unit (CPU).



APB = Advanced Peripheral Bus AHB = Advanced High-performance Bus

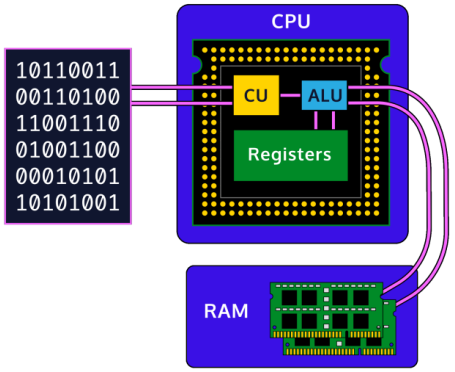
ARM Cortex includes Debug Port ➡ IDE

- JTAG (Joint Test Action Group) specifies the use of a dedicated debug port implementing a serial communications interface for low-overhead access without requiring direct external access to the system address and data buses.

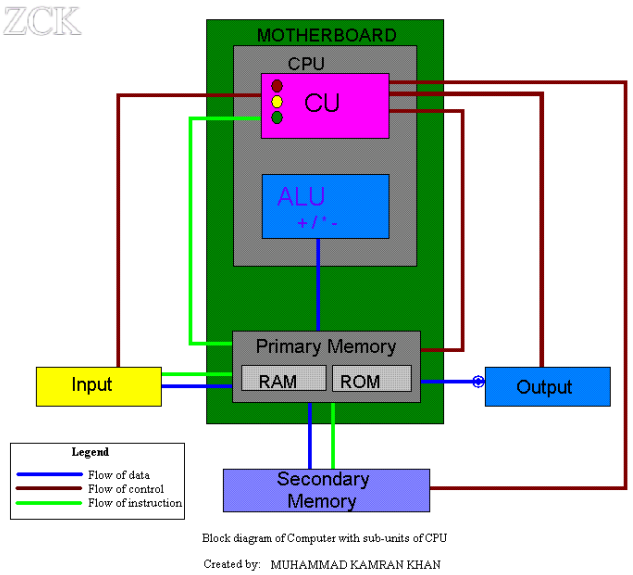


APB = Advanced Peripheral Bus AHB = Advanced High-performance Bus

Outline



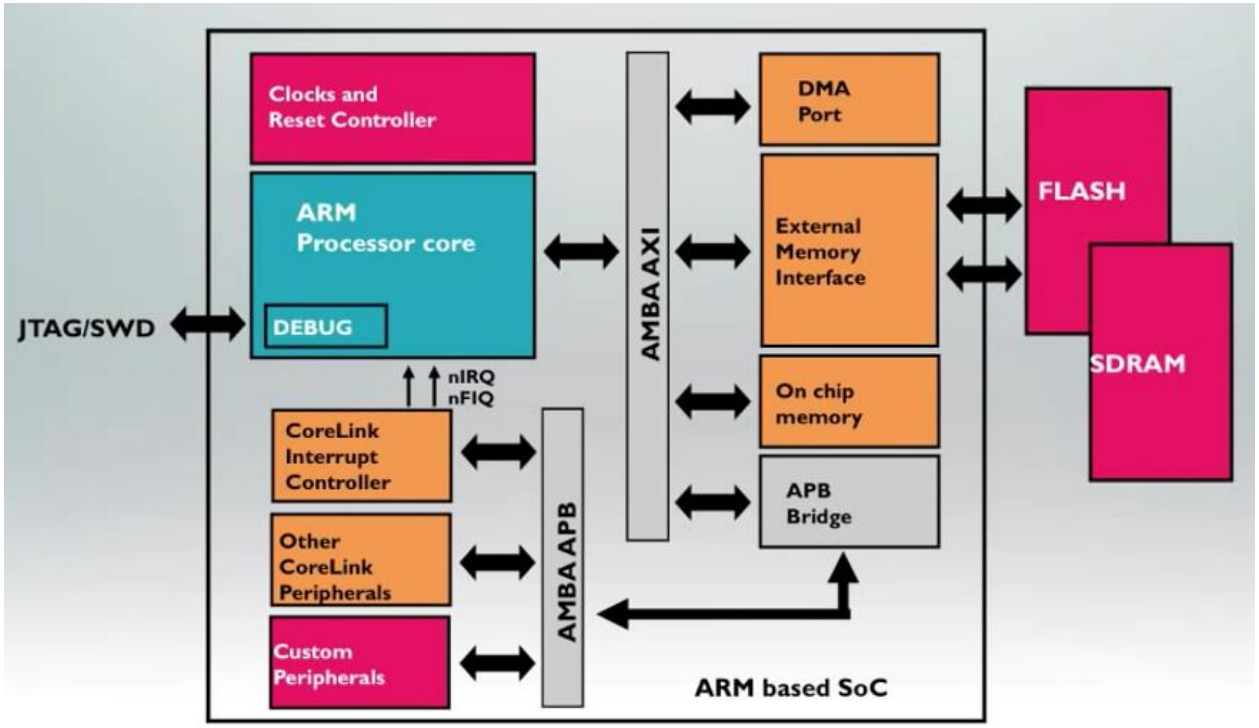
- Basics of Brains
- Inventors of Digital Computers
- Family of ARM Microprocessors



- Integrated Development Environment

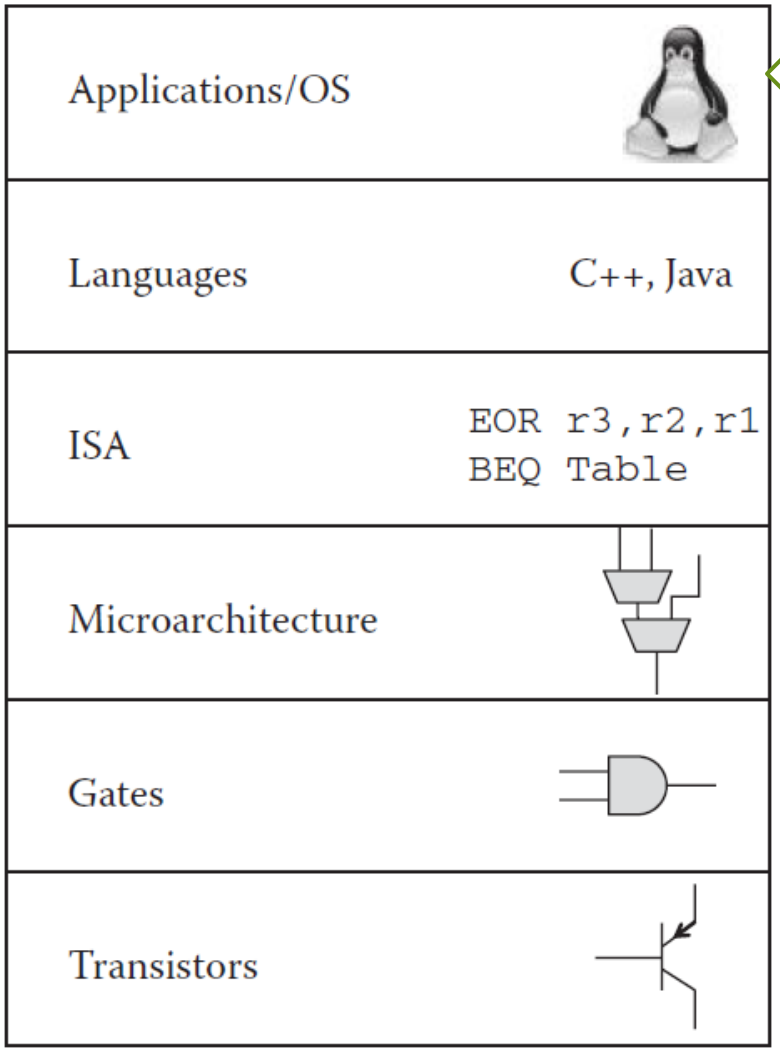
What is IDE

- IDE stands for integration development environment which consists of a graphical user interface and a set of software tools so as to facilitate users to translate their solutions into computational flows to be executed by a microcontroller.



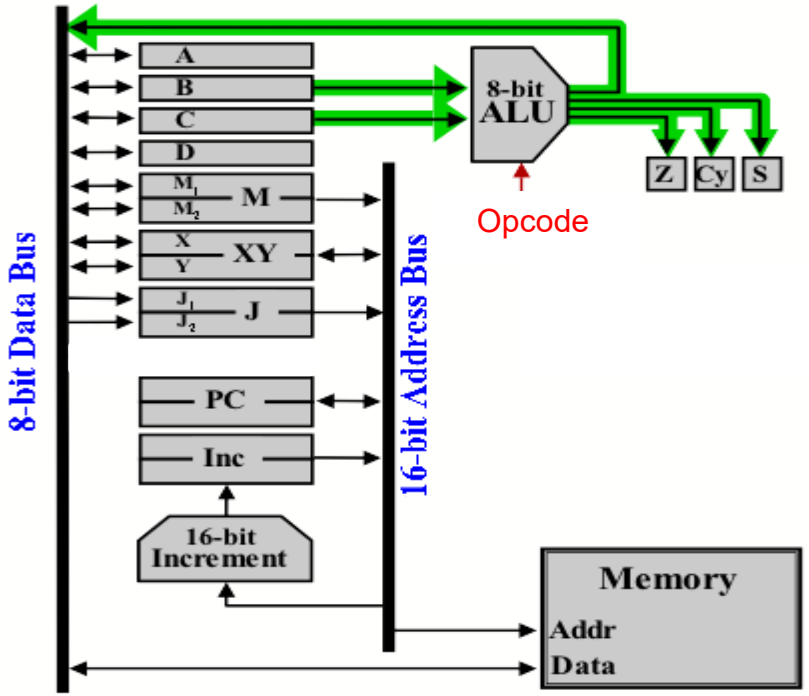
APB = Advanced Peripheral Bus AHB = Advanced High-performance Bus

Landscape of Software Development

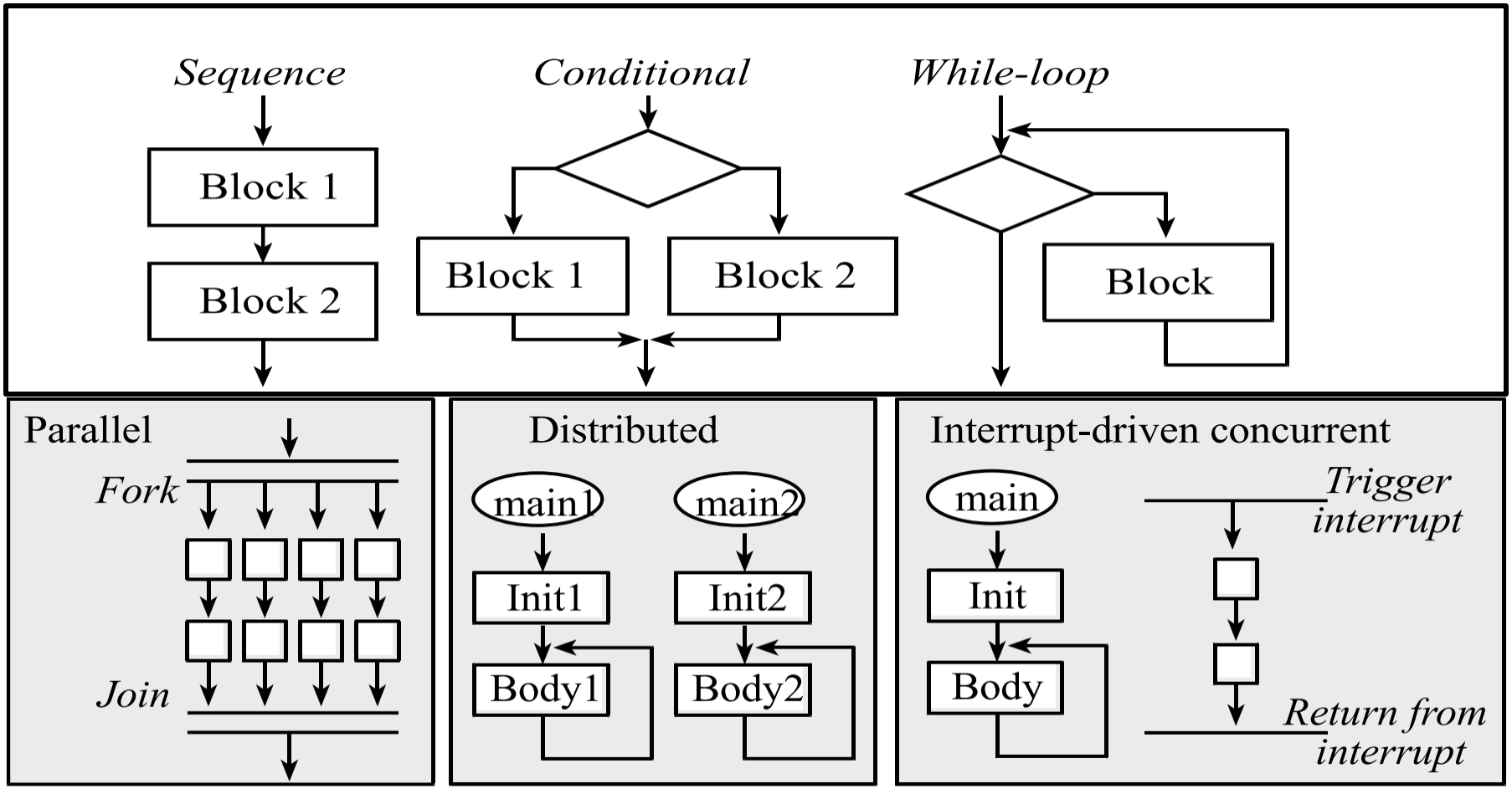


Algorithms or Solutions

Problems to be solved

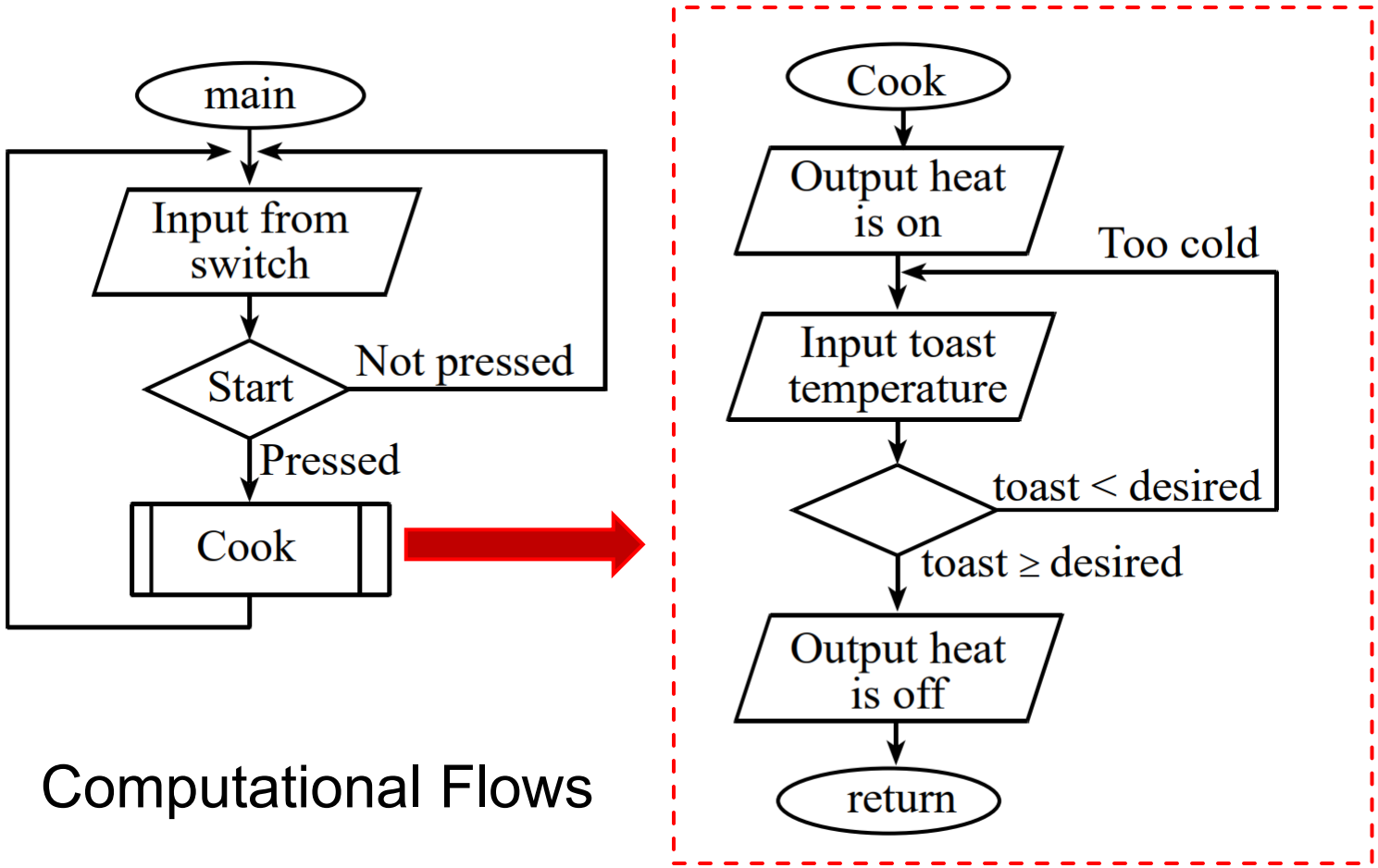


Typical Types of Application Software



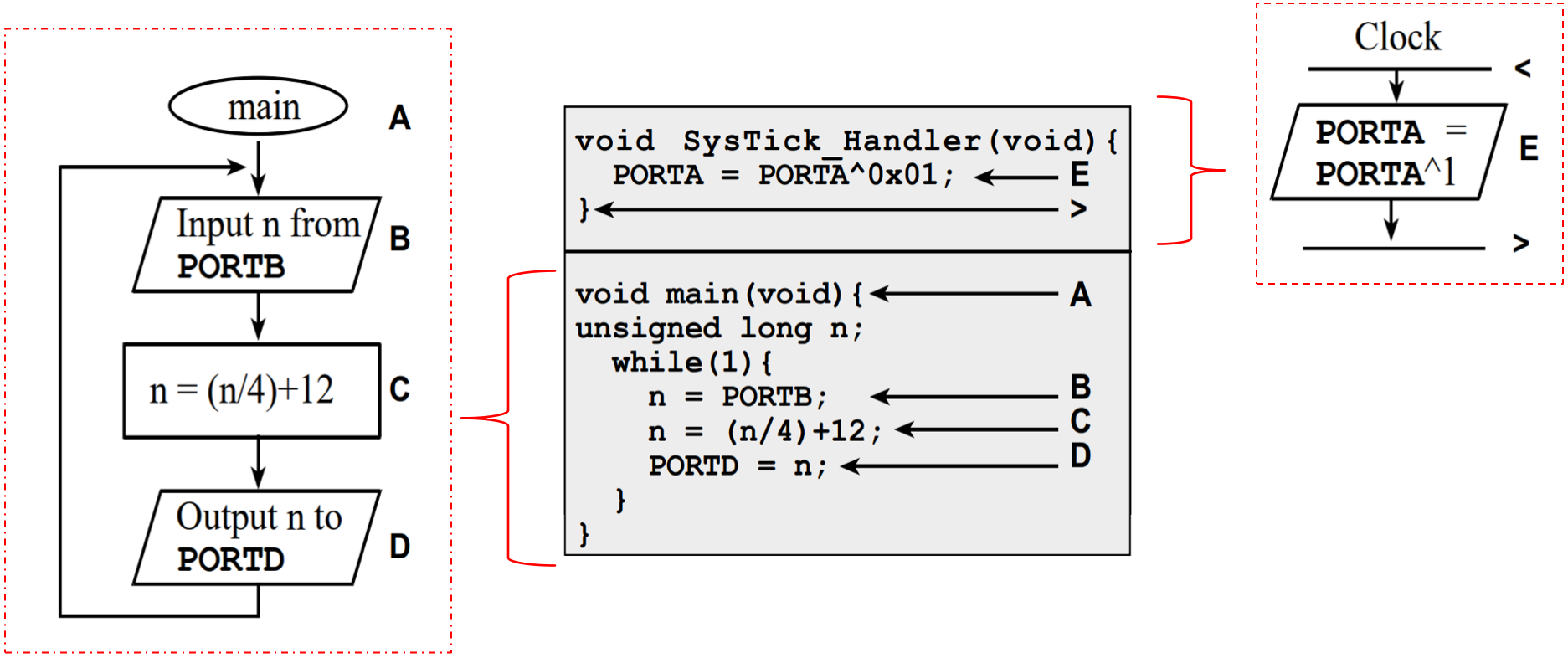
From Algorithms to Flowcharts

Example of Solutions in an Electric Toaster



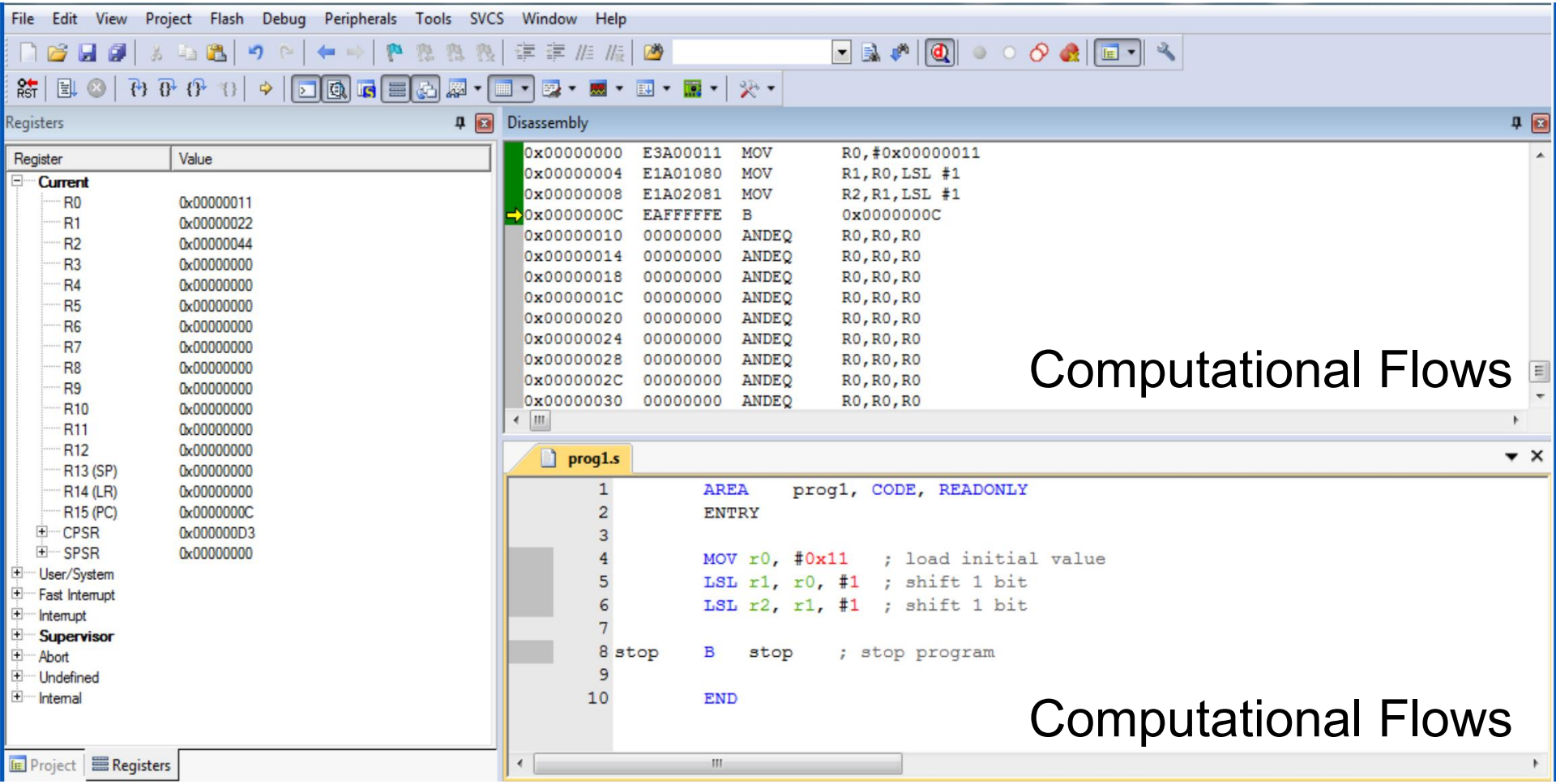
Computational Flows

From Flowcharts to Programs

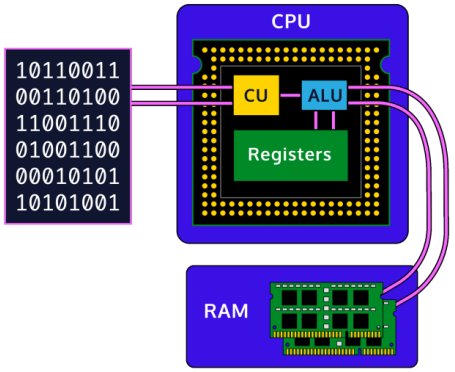


Computational Flows

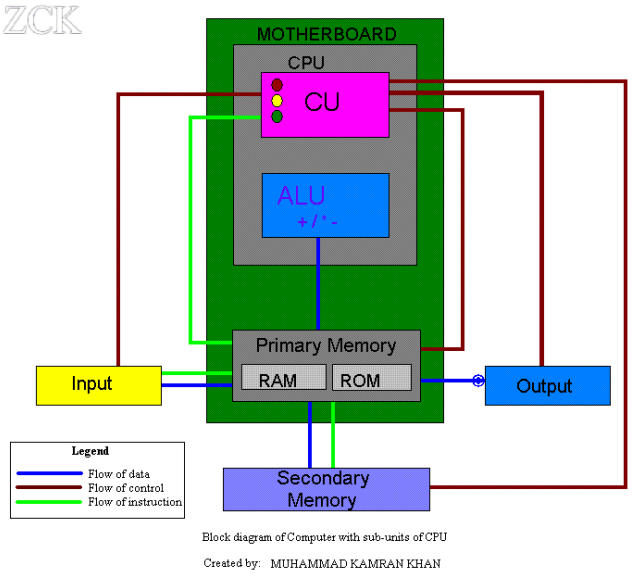
From Programs to Executions



Summary



- Basics of Brains
- Inventors of Digital Computers
- Family of ARM Microprocessors
- Integrated Development Environment



"Ask not what your country can do for you – ask what you can do for your country," - John F. Kennedy

"Do not think that you are needy – think that you are needed in the world", - Manis Friedman

"Study will make you knowledgeable, resourceful, and hence more needed", - Xie Ming

Thank You for Listening!