

Welcome to the CET 241: Introduction to Embedded Microcontroller Systems

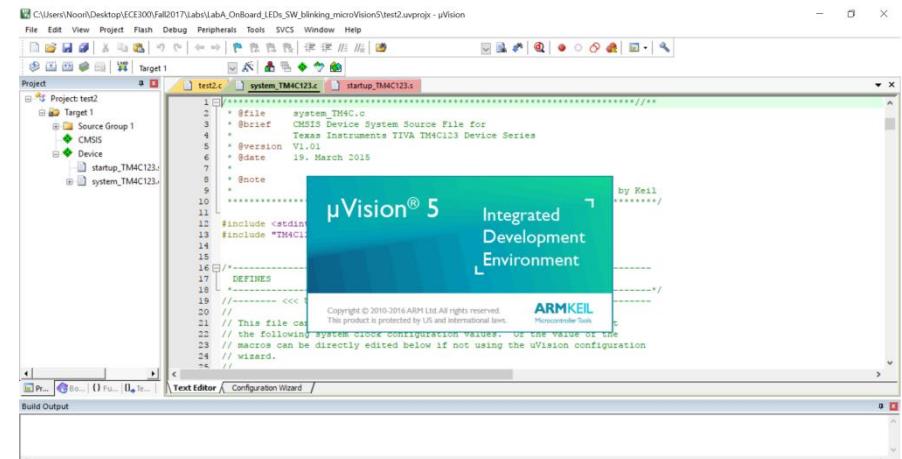
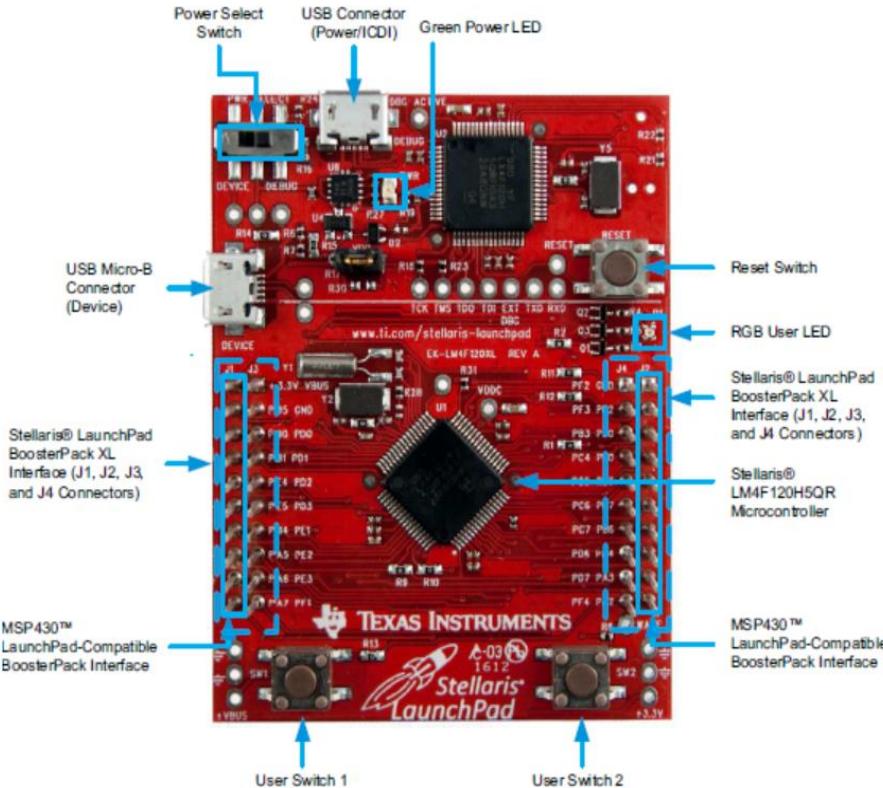
Dr. Noori Kim

About this course

- Runs for 18 days
 - Attendance is mandatory send me an absence email
 - CET241_your name: ‘Mr./Ms.Late’ absence '(DD/MM/YYYY)’
 - Content i.e., “I woke up in a bigger house”
- Grading:
 - 40%: 10 online quizzes, each 4%
 - 60%: 4 offline quizzes or demos, each 15%.
 - No makeup exams or quizzes
- My office hours/contact info
 - Appointment via email/phone
 - noori.kim@digipen.edu, #1743

- Textbook
 - Embedded Systems: Introduction to Arm® Cortex-M Microcontrollers, Volume 1, by Johnathan W. Valvano (ISBN 9781477508992)
- Recommended book
 - Embedded Systems: Real-Time Interfacing to ARM® Cortex™-M Mictocontrollers, Volume 2, by Johnathan W. Valvano (ISBN 9781463590154)

- Equipment/Software Required
 - Texas Instruments Tiva C TM4C123G LaunchPad board
 - Keil uVision V5



Academic integrity: Any form of dishonesty will not be tolerated in this course

– Value yourself, value your integrity

Useful resources

- Author's book on C programming (Embedded Software in C for an ARM Cortex M):
<http://users.ece.utexas.edu/~valvano/embed/toc1.htm>
- Author's website (John Valvano, The University of Texas: where this course was developed): pros and cons => extensive (too much) resources
 - <http://users.ece.utexas.edu/~valvano/Volume1/>
 - <http://users.ece.utexas.edu/~valvano/arm/lectures2.html>

OK, let's get started ☺

What is a system?

system

/'sɪstəm/ 

noun

plural noun: **systems**

1. a set of things working together as parts of a mechanism or an interconnecting network; a complex whole.
"the state railway system"
synonyms: [structure](#), [organization](#), [order](#), [arrangement](#), [complex](#), [apparatus](#), [network](#); [More](#)
2. a set of principles or procedures according to which something is done; an organized scheme or method.
"a multiparty system of government"
synonyms: [method](#), [methodology](#), [technique](#), [process](#), [procedure](#), [approach](#), [practice](#), [line](#), [line of action](#), [line of attack](#), [attack](#), [means](#), [way](#), [manner](#), [mode](#), [framework](#), [modus operandi](#); [More](#)



Translations, word origin, and more definitions

System

From Wikipedia, the free encyclopedia

For other uses, see [System \(disambiguation\)](#).

A **system** is a set of [interacting](#) or interdependent component parts forming a complex/intricate whole.^[1]

A system example: watch



- A time display SYSTEM
- Parts: hardware, needles, battery, dial, chassis and strap
- (**Interconnecting/inter-dependent**) Rules
 - All needles move clockwise only
 - A thin needle rotates every second
 - A long needle rotates every minute
 - A short needle rotates every hour
 - All needles return to the original position after 12 hours



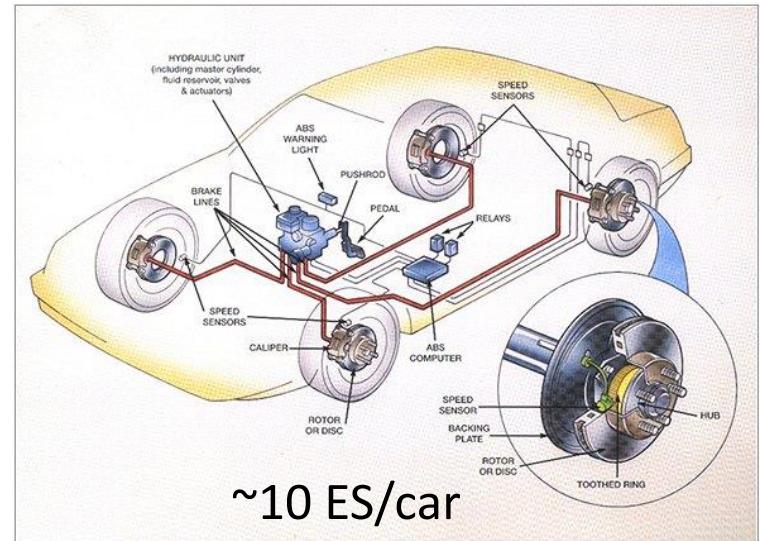
Embedded systems?

- “Embedded”: **built into**
 - **Software** is built into hardware → Computer systems
- **A computer system with a dedicated function**
 - A permanent part for solving **only a few very specific problems**, not easily changed *by Wikipedia*

- Examples of Embedded Systems (ES): a pacemaker, an iPod, a digital watch, and etc...
- A general-purpose computer *vs.* ES



VS.



- does many different jobs
- can be changed at any time for new jobs.

Typical Embedded System Constraints

- Small Size, Low Weight: handheld electronics
- Low Power
 - Battery power for 8h to 10y
 - Limited cooling, limit power although AC power available
- Harsh environment
 - Heat, vibration, shock, power fluctuations, RF interference, lightning, water, corrosion, physical abuse
- Safety critical operation
- Extreme cost sensitivity

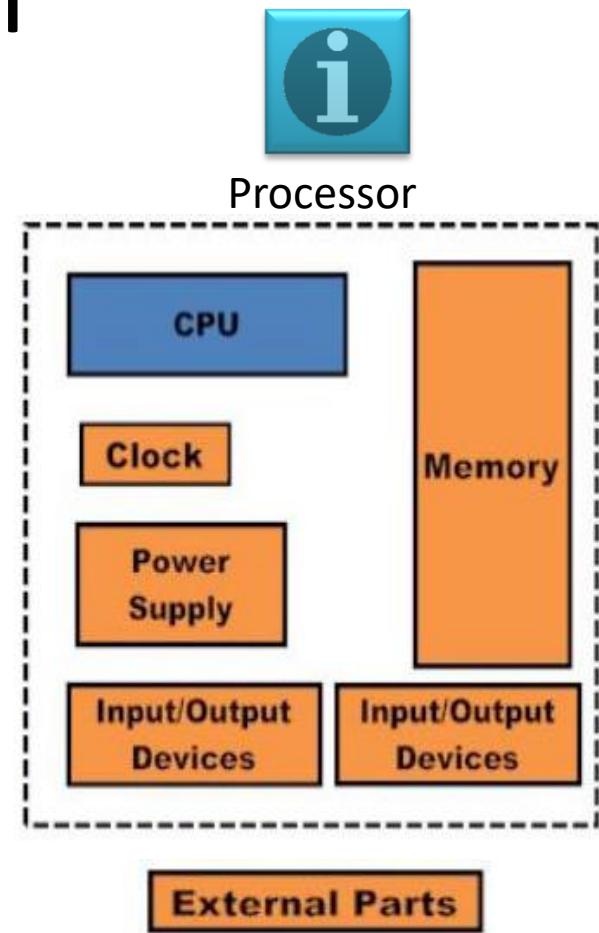
Embedded microcontroller systems?

- **Modern embedded systems** are often based on **microcontrollers** (i.e. CPUs with integrated memory or peripheral interfaces)... by Wikipedia
- A microcontroller is used as a brain of Embedded systems

Then, what is the microcontroller?
Differences compared to a microprocessor?

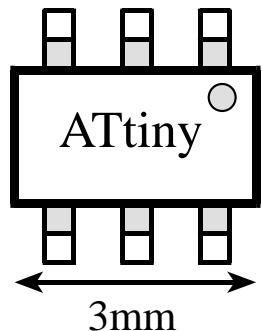
A processor

- Executes software
- Needs other external parts to make a workable computer
- A **microprocessor??**
 - Micro means small, i.e., 10^{-6}
 - Just a small _____
 - Micro-, nano- technologies
- Ex: I7 etc.



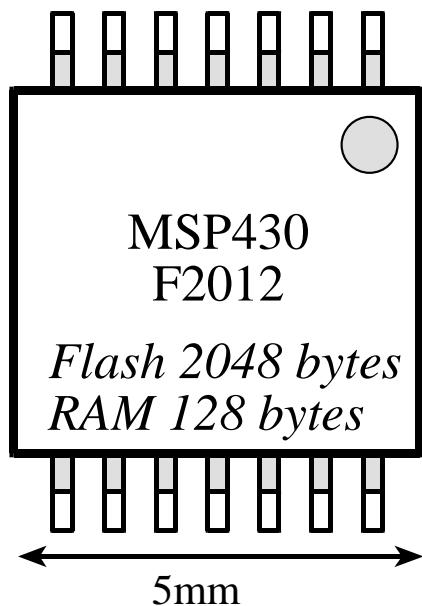
A microcontroller?

- A single chip computer

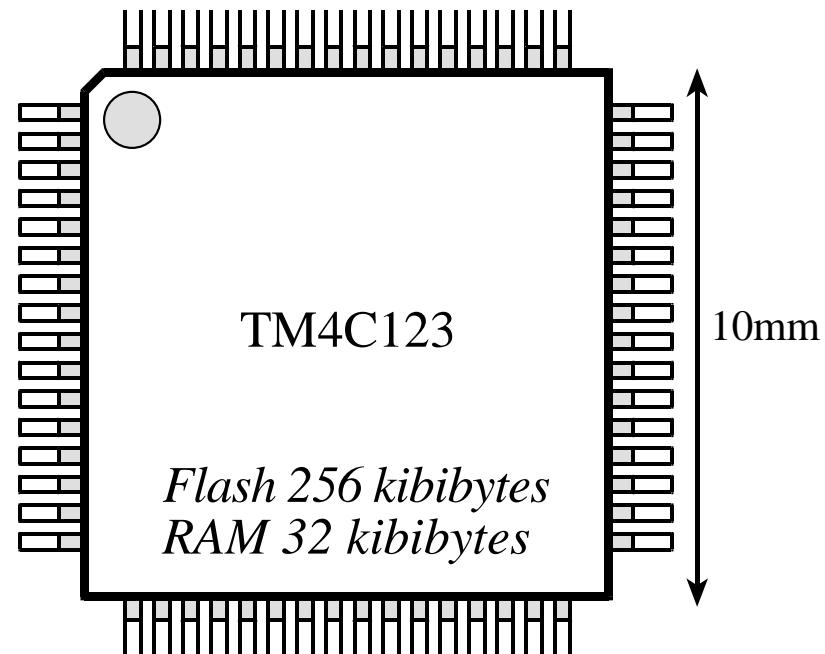


*Flash 512 bytes
RAM 32 bytes*

Atmel

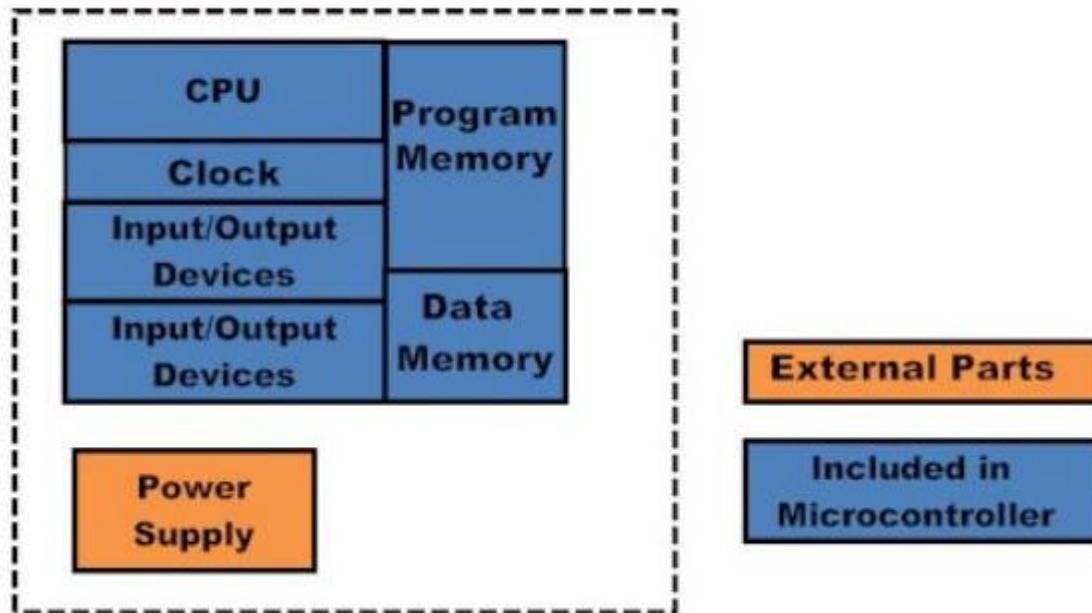


Texas Instrument

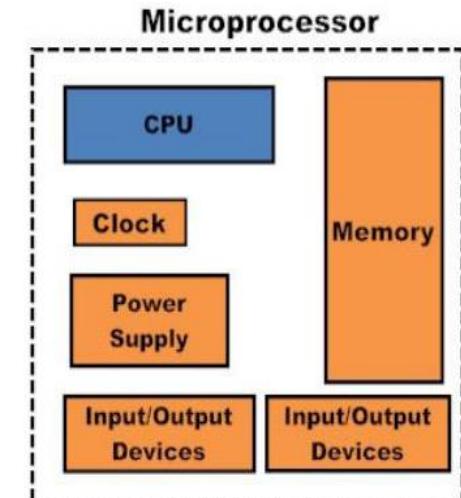


Texas Instrument
(Our MCU)

Microcontroller



- Recall that



Microprocessors (MP) vs. Microcontroller (MC)

The functional blocks: ALU, registers, timing & control units	Functional blocks: microprocessors & in addition timer, parallel I/O, RAM, EPROM, ADC & DAC
Bit handling instruction: one or two type only	Many type of bit handling instruction
Rapid movements of code and data between external memory & MP	Rapid movements of code and data within MC
Used for designing general purpose digital computers	Used for designing specifically dedicated application

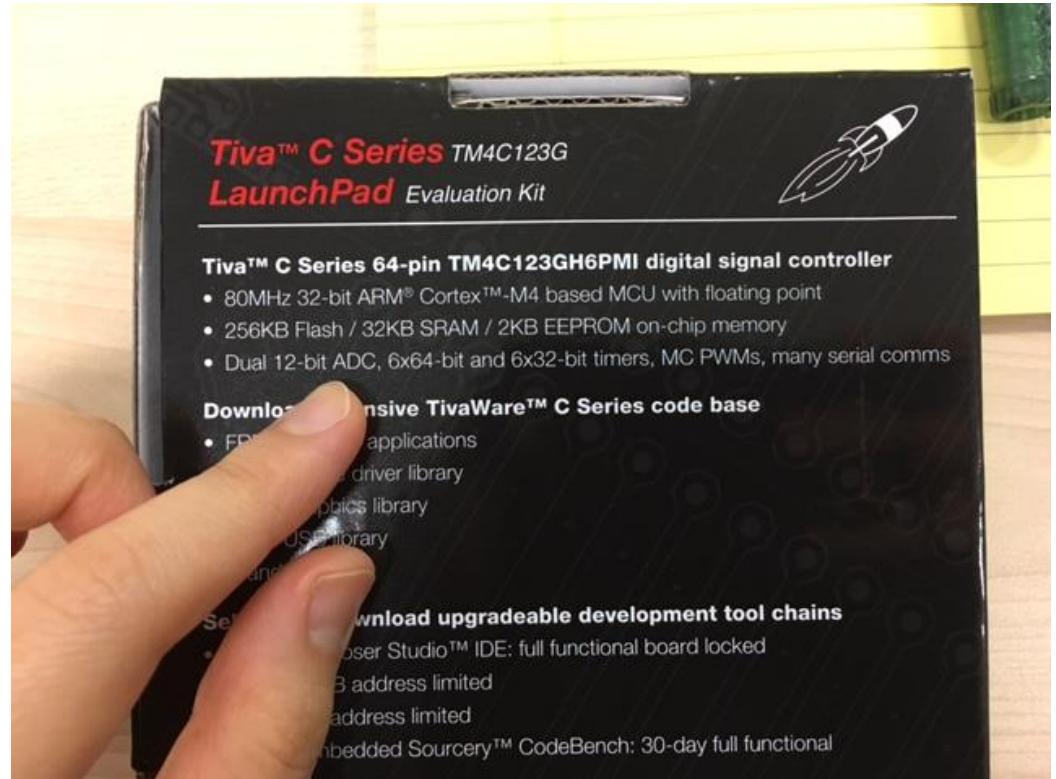
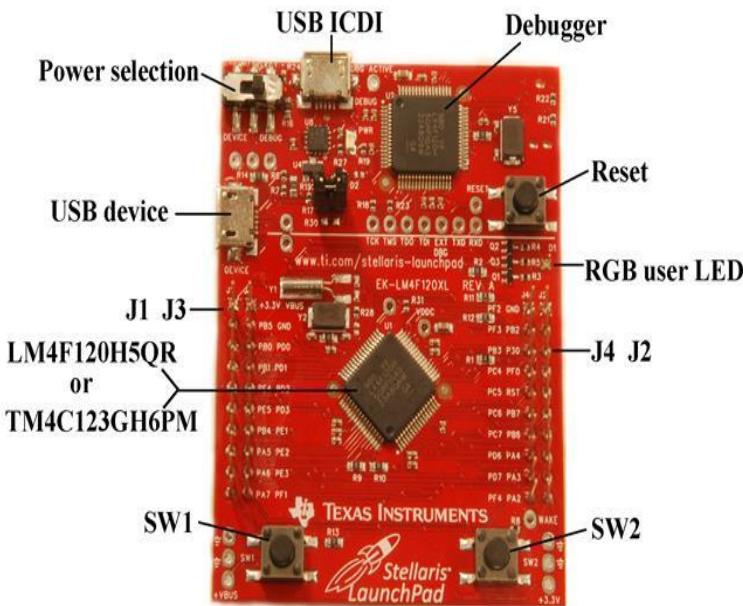
Intro to the ARM Cortex M processor

Dr. Noori Kim

Agenda

1. ARM history
2. ARM architecture
 - ARM ISA: CISC or RISC?
 - CPU, Bus, Memory, I/O ports: Harvard or von Neumann?
 - AMBA?
 - Registers, Memory (map, little/big endian)
3. TM4C123 MCU: Type of I/O, Memory map

Tiva™ C Series LaunchPad



- 64-pin TM4C123GH6PM digital signal controller
- Our MCU board has an MCU
 - 80MHz/32bit/ **ARM® CortexTM –M4** based MCU
 - 256KB Flash/32KB SRAM/2KB EEPROM

[16-bit Microcontrollers - MCU \(9,591\)](#)[32-bit Microcontrollers - MCU \(4,168\)](#)[8-bit Microcontrollers - MCU \(15,425\)](#)[ARM Microcontrollers - MCU \(7,593\)](#)

- Kinds of MCU: 9k+15k+7k+4k....=35k....
 - Available in Singapore.. Then in the world??
 - How many kinds of board will be available???
 - One can make diff. boards using the same MCU
- An expert:
 - knows general pattern/features of the device
 - Can estimate the time that she/he would spend to complete a given task using MCU

Then specified features that I am looking for

Category Semiconductors > Embedded Processors & Controllers > Microcontrollers - MCU > ARM Microcontrollers - MCU

Available Filters 7,593 Matches 72 remaining
 Enable Smart Filtering [?](#)

Manufacturer	Mounting Style	Package / Case	Core	Data Bus Width	Maximum Clock Frequency	Program Memory Size	Data RAM Size	ADC Resolution
Analog Devices Inc.	SMD/SMT	LQFP	ARM Cortex A5	32 bit	32 kHz	0 kB	1 kB	1.8 V to 5.5 V
Atmel	Through Hole		ARM Cortex A5, ARM Cortex M4	32 bit/16 bit	10 MHz	4 kB	2 kB	4 x 12 bit, 1 >
Cypress Semiconductor			ARM Cortex M0		14.7456 MHz	8 kB	4 kB	6 bit, 12 bit
GHI Electronics			ARM Cortex M0+		16 MHz	16 kB	4096 B	8 bit
IDT (Integrated Device Technology)			ARM Cortex M0, ARM Cortex M4		20 MHz	24 kB	6 kB	10 bit
Infinion			ARM Cortex M3		20.48 MHz	32 kB	6144 B	10 bit
Maxim Integrated			ARM Cortex M3, TMS320C28x		24 MHz	40 kB	8 kB	10 bit, 12 bit
Microchip			ARM Cortex M4		25 MHz	48 kB	8192 B	10 bit/12 bit
NXP			ARM Cortex M4, ARM Cortex M0		28 MHz	56 kB	10 kB	12 bit
ON Semiconductor			ARM Cortex M4, ARM Cortex M0+		30 MHz	62 kB	12 kB	12 bit, 16 bit
Silicon Laboratories			ARM Cortex M4F		32 MHz	64 kB	16 kB	13 bit
STMicroelectronics			ARM Cortex M7		33 MHz	80 kB	16 kB, 20 kB	14 bit
Texas Instruments			ARM Cortex R4		36 MHz	88 kB	18 kB	15 bit
Toshiba			ARM Cortex R4F		40 MHz	96 kB	20 kB	16 bit
VORAGO Technologies			ARM7TDMI-S		40.8 MHz	126 kB	24 kB	16 bit, 24 bit
WIZnet			ARM7TDMI-S		41.78 MHz	128 kB	32 kB	17 bit
ZILOG			ARM9		45 MHz	160 kB	33 kB	24 bit
			ARM920T		48 MHz	192 kB	34 kB	No ADC
			ARM926EJ-S		50 MHz	200 kB	35 kB	
			ARM966E-S		55 MHz	256 kB	36 kB	

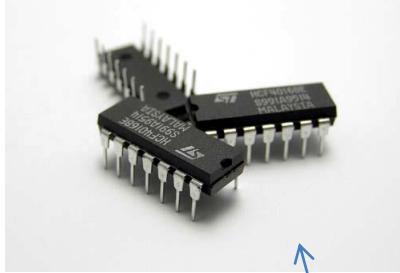
Reset [Apply Filters](#)

Mounting style, package/case?

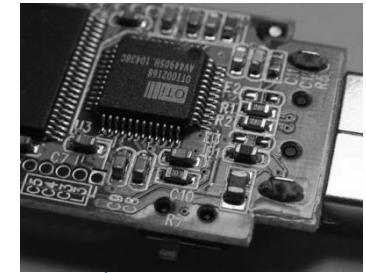


80MHz 32bit ARM® Cortex™-M4 based MCU
256KB Flash/32KB SRAM/2KB EEPROM

Pins are drilled through PCB



Mounted via leads on PCB



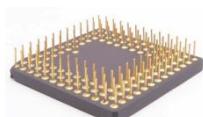
MCU packaging types

Thru-Hole Device (THD)

SDIP (Shrink DIP)

PGA (Pin Grid Array)

Etc...

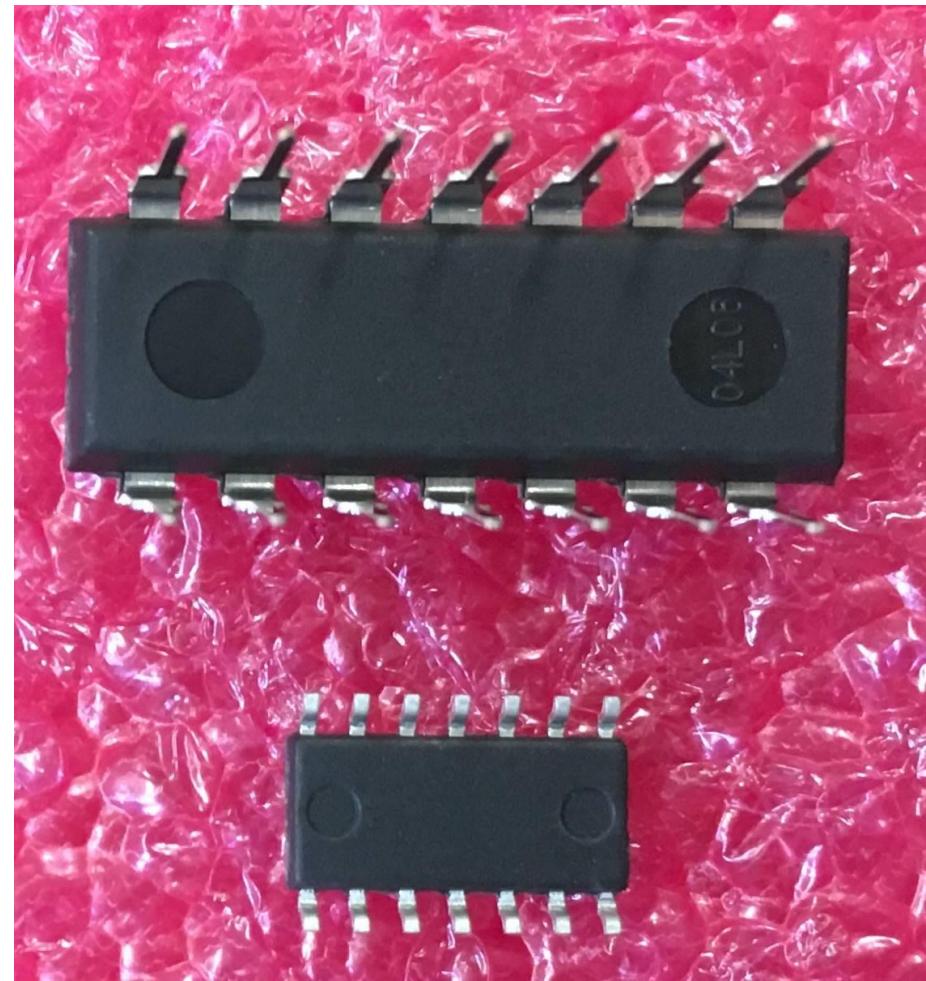
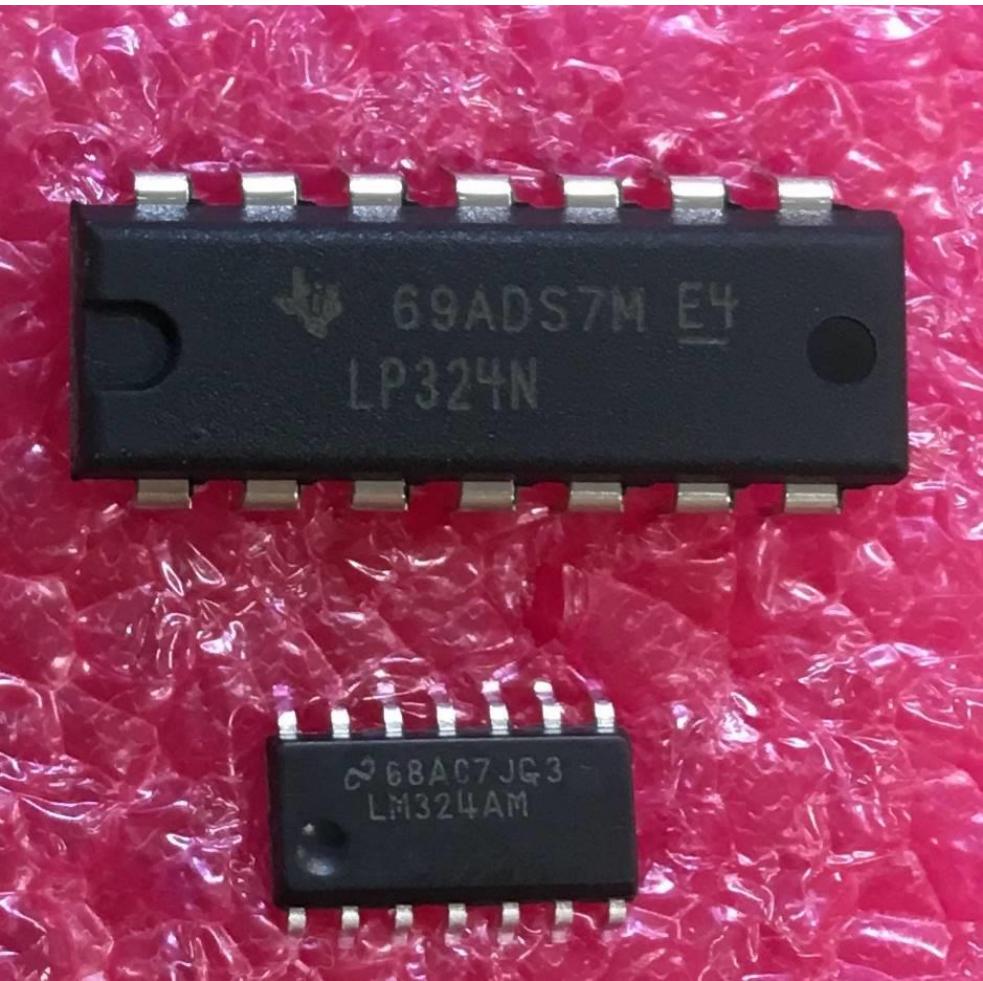


Surface-Mount Device (SMD)

LQFP (Low Profile Quad Flat Pack)

Etc...

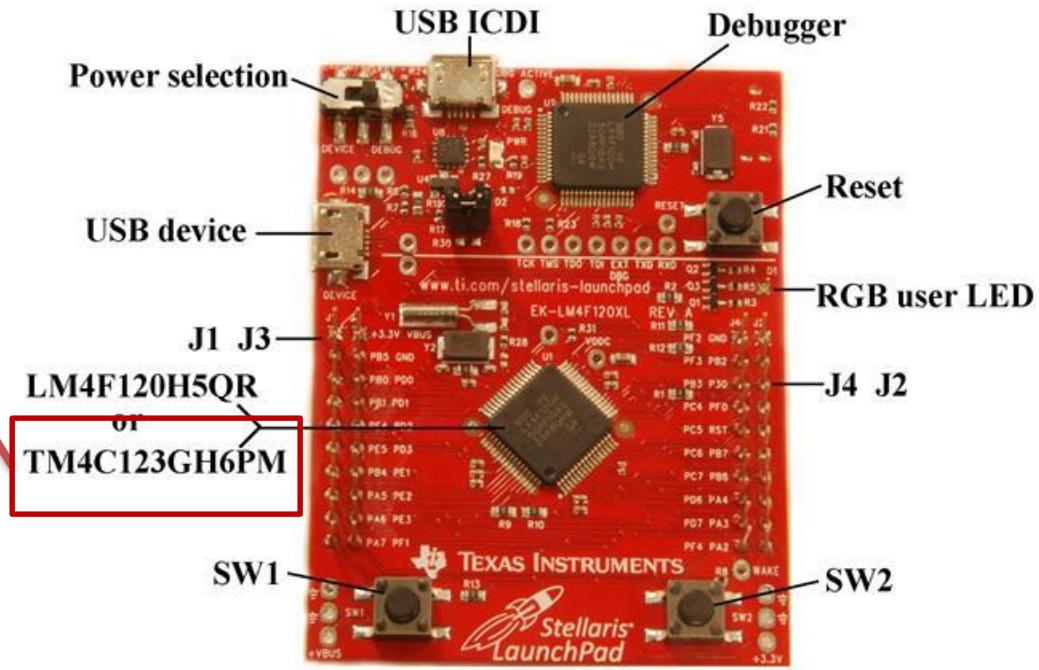




Hear we go~!

Image	Mouser Part #	Mfr. Part #	Mfr.	Description	Datasheet	Availability	Pricing (SGD)	Quantity	RoHS	Mounting Style	Package / Case	Core	Data Bus Width	Maximum Clock Frequency	Program Memory Size	Data
	595-TM4C123GH6PMI	TM4C123GH6PMI	Texas Instruments	ARM Microcontrollers - MCU Tiva C Series MCU Learn More	Datasheet	931 In Stock Alternative Packaging	1: \$17.59 10: \$16.17 25: \$15.32 50: \$14.91 100: View	<input type="button" value="Buy"/> Min.: 1 Mult.: 1	 Details	SMD/SMT	LQFP-64	ARM Cortex M4	32 bit	80 MHz	256 kB	32

[Enlarge](#)



- We find our MCUs in markets.
- Take-home messages
 - The ARM is a kind/core of MCUs (in this context)
 - Our Lanuchpad uses an **MCU (TM4C123GH6PM**, black, size: 1cm²) which has
 - ARM-Cortex M4 as its core
 - 256KB Flash/32KB SRAM/2KB EEPROM
 - Up to 80-MHz operation
 - 64 pins
 - Our **Launchpad (EK-TM4C123GXL)** is an evaluation board which includes many physical connections/components such as
 - LEDs, switches, 5 V battery connectors, On-board debugger, USB Micro-AB connector, resistors, capacitors, crystals.....

ARM Cortex-M4?



ARM Cortex-M

From Wikipedia, the free encyclopedia

The **ARM Cortex-M** is a group of **32-bit RISC ARM** processor cores licensed by **ARM Holdings**. The cores are intended for **microcontroller** use, and consist of the Cortex-M0, M0+, M1, M3, M4, and M7.^{[1][2][3][4][5]}

Later

Announced	
Year	Core
2004	Cortex-M3
2007	Cortex-M1
2009	Cortex-M0
2010	Cortex-M4
2012	Cortex-M0+
2014	Cortex-M7

Our MCU board: an ARM®Cortex™ -
M4 based MCU
– 7 years old tech.

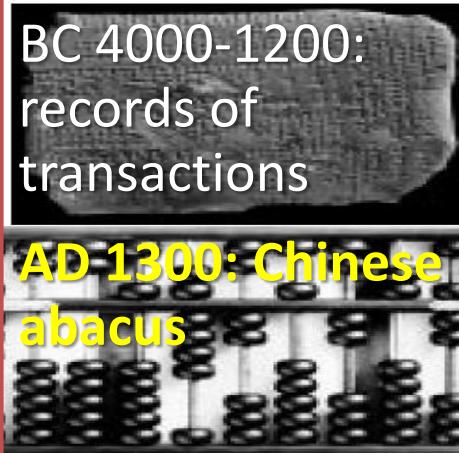


Over-simplified history of ARM

Computing

BC 4000-1200:
records of
transactions

**AD 1300: Chinese
abacus**



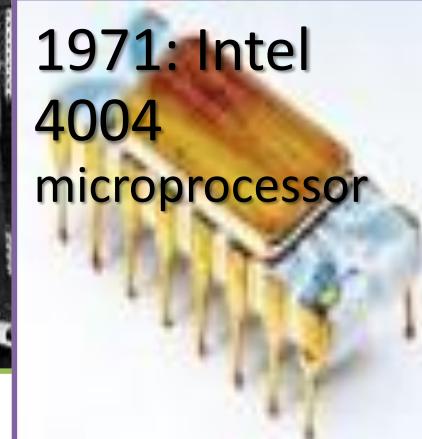
Computer

1946: ENIAC
(Electronic
Numerical Integrator
And Computer)
the United
States Army



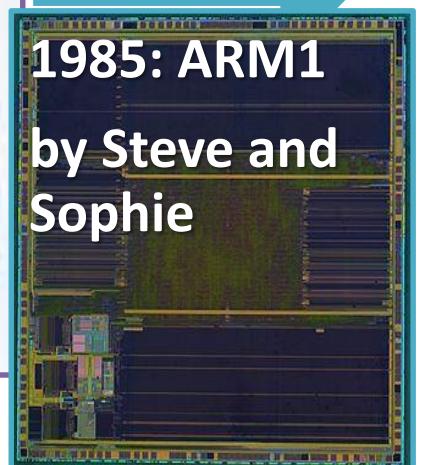
Microprocessor

1971: Intel
4004
microprocessor



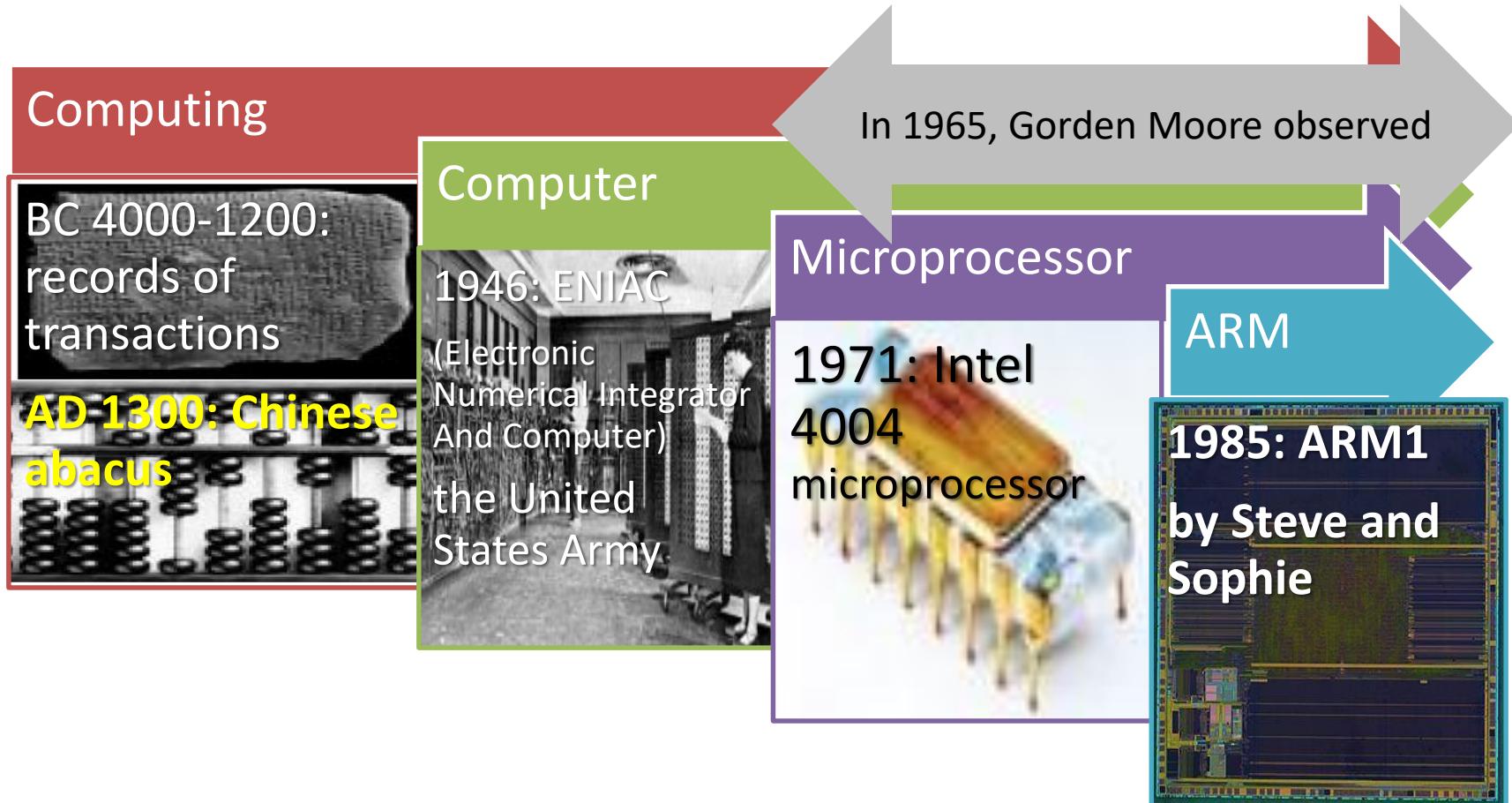
ARM

1985: ARM1
by Steve and
Sophie



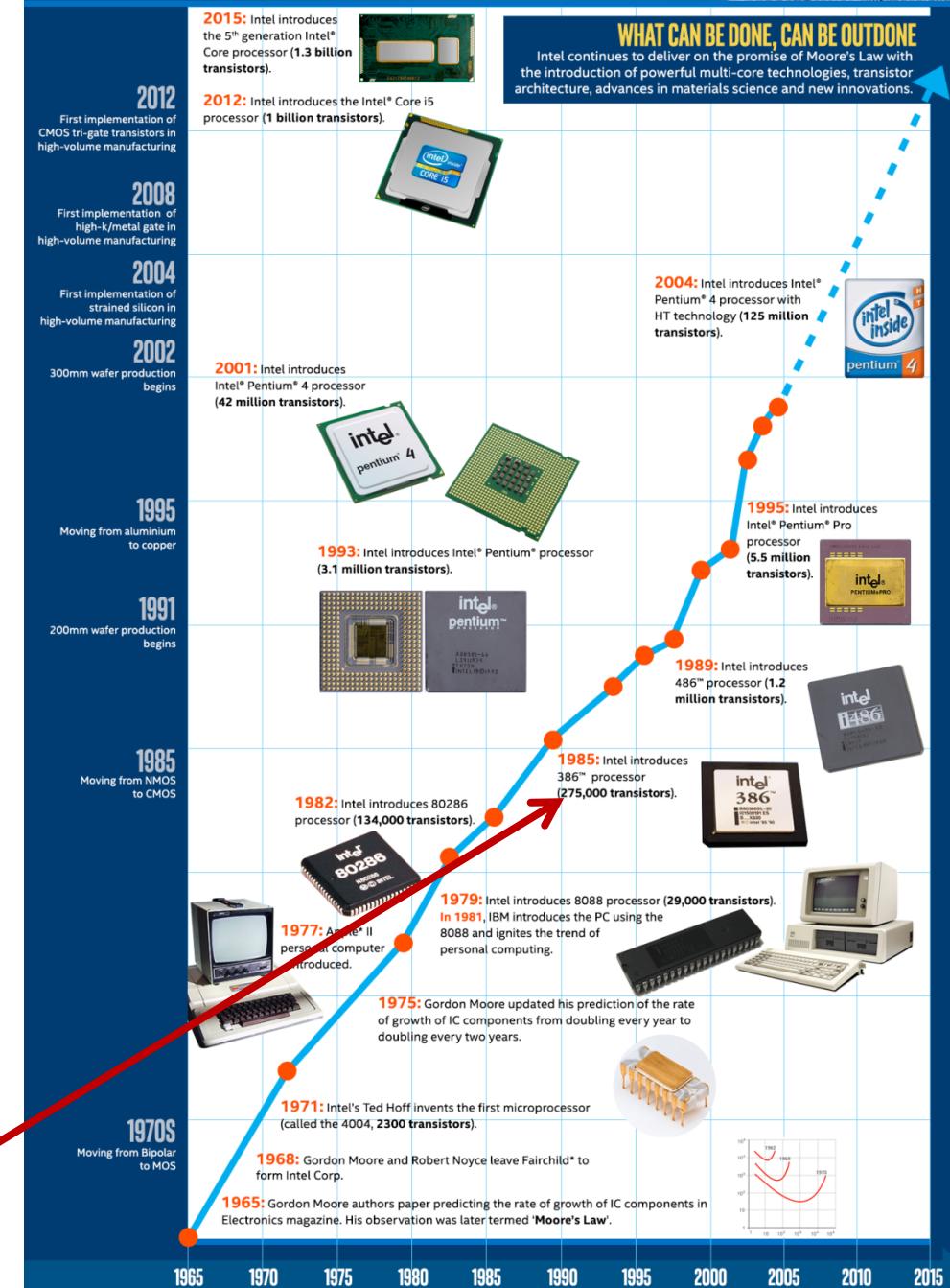
[A supplement - history of computer. pdf](#)

Over-simplified pre-history of ARM



Moore's law

- # of transistors per inch² on integrated circuits -> doubled every year since 1965
- Size smaller, speed faster



For more information, please visit intel.com.

Intel, Pentium, Core, Intel386, Intel486, the Intel logo and the Intel Inside logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. *Other names and brands may be claimed as the property of others.

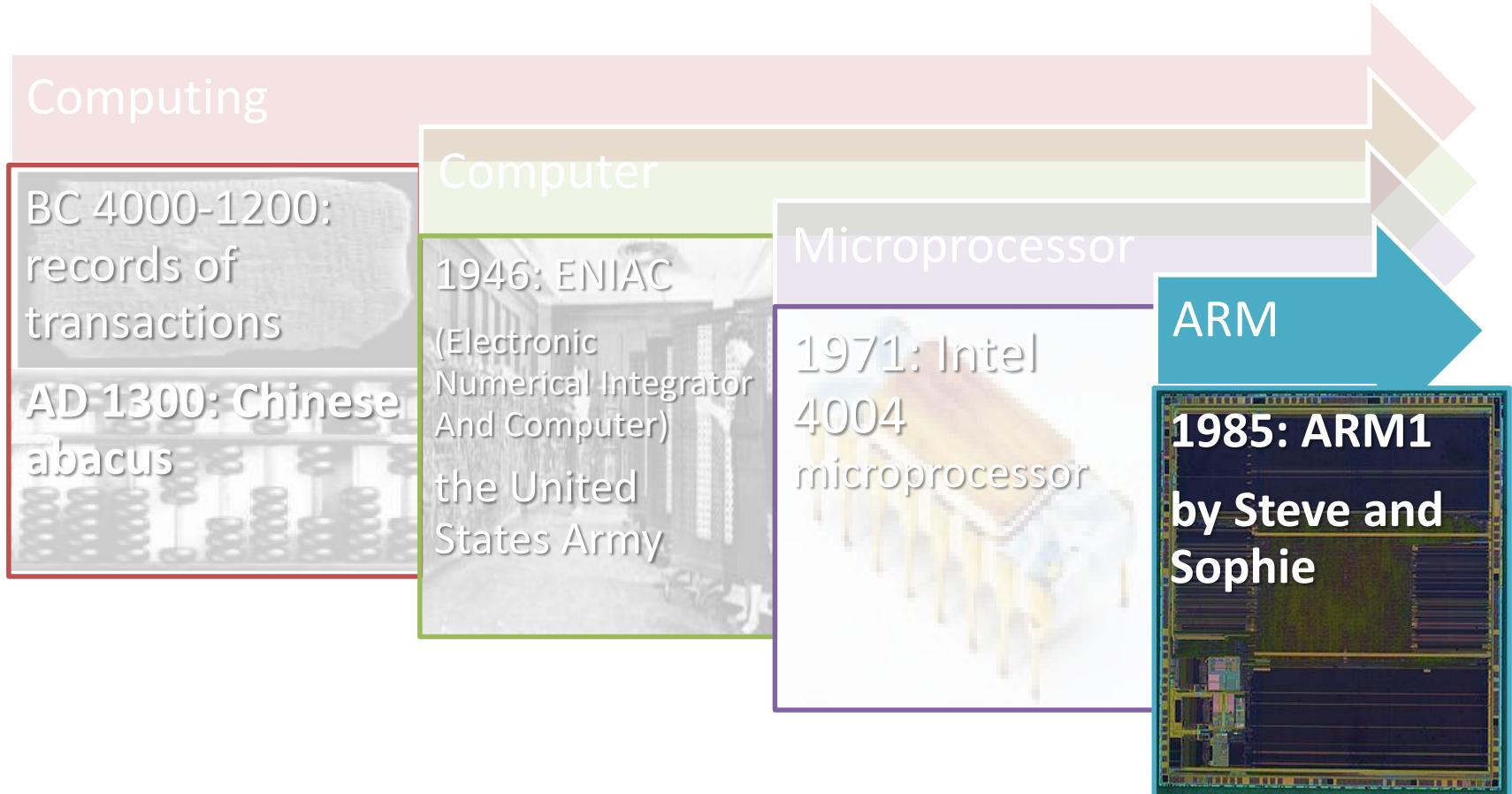
A note

- Special about 2015
 - 50th anniversary of Moore's law (since 1965)
 - 25th anniversary of ARM1 (since 1985)
- Useful information has been released from the companies to celebrate their anniversaries.

A message from Gorden Moore (2min)



Over-simplified pre-history of ARM



- ARM: Acorn RISC (Reduced Instruction Set Computing) Machine => A RISC Machine
- ARM1: Working samples were received in 1985
- What makes it special?
 - Low cost/power

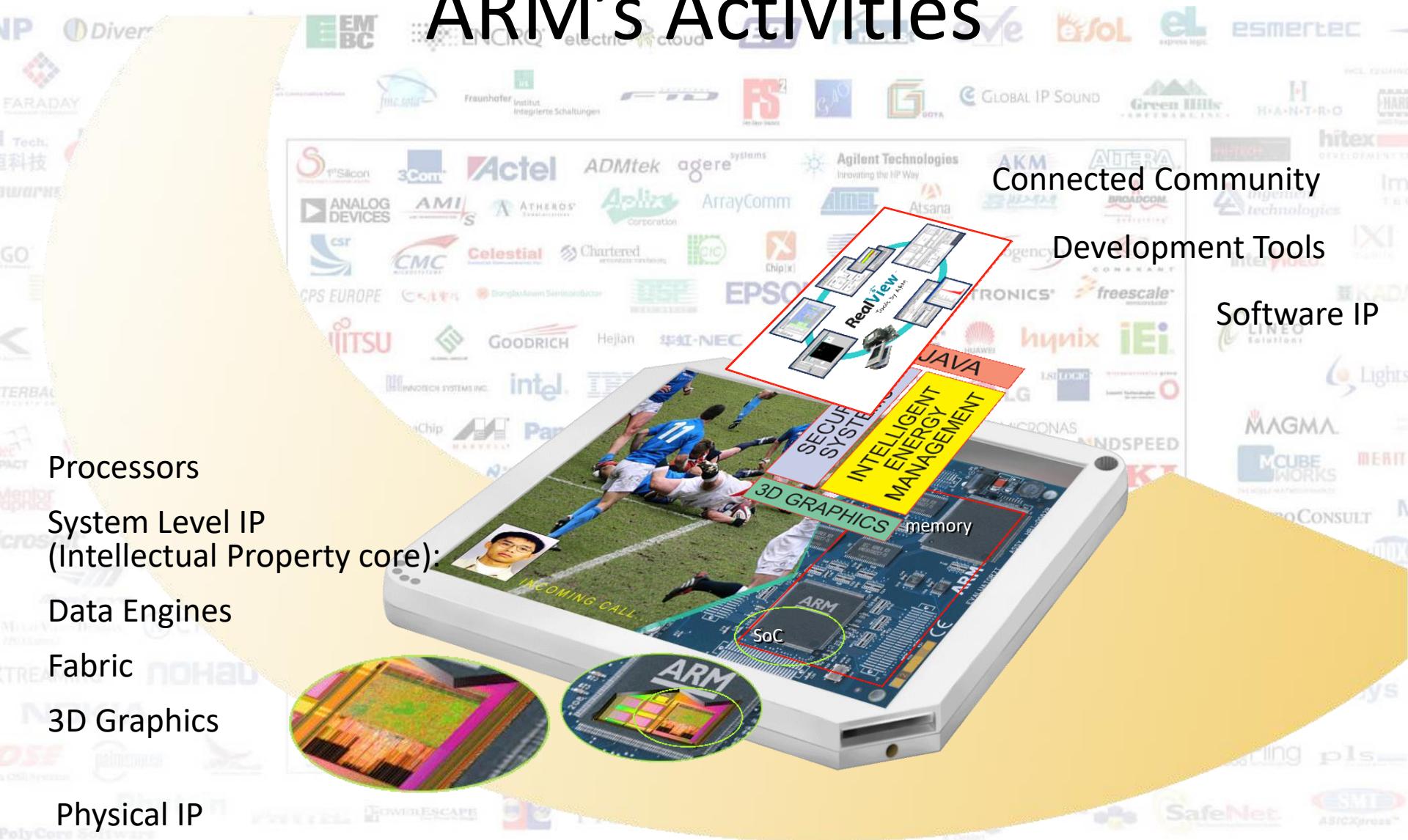


About the company: ARM Ltd

- Founded in November 1990
 - Spun out of Acorn Computers
 - Initial funding from Apple, Acorn and VLSI
- ARM Ltd does not fabricate silicon
 - Licenses ARM core designs to semiconductor partners who fabricate and sell to their customers



ARM's Activities



ARM Connected Community – 700+



An extensive 3rd party community:
to make the whole device creation possible



Birth of the ARM1 processor



By Steve Furber one of two engineers based at Acorn Computers in Cambridge, UK (13min)

The ARM architecture

- Computers are electronic i _____
 - Store a lot of data with zeros and ones
 - Only do exactly what we tell them to do
 - Don't get b _____ doing the same tasks
- But they can execute our order quite q _____

Check!

- When following terms used in relation to ARM cores:
 - Byte means 8 bits
 - Halfword means 16 bits (2 bytes)
 - Word means 32 bits (4 bytes)

The computer architecture

■ Instruction Set Architecture: ISA

- RISC (Reduced instruction set computer)
- CISC (Complex instruction set computer)

□ Processor design

- Von Neumann architecture
 - Harvard architecture
-
- ARM architecture, point to the answer?
 - RISC and Harvard

Instruction Set Architecture: ISA (1/3)

Machine language	Assembly language
Encoding 0's and 1's: binary	Writing in textual form
Copy the value from "Register 9" into "Register 3."	
1110 0001 1010 0000 0011 0000 0000 1001	MOV R3, R9

- ISA: The **design** of the **machine language encoding**
- (Wiki link) List of instruction sets:

[List of instruction sets](#)

From Wikipedia, the free encyclopedia

A list of [computer](#) central processor **instruction sets**: (Companies that created only a few different processors are listed at the end under "Other")

Contents [hide]
1 AMD
2 Analog Devices
3 ARM
4 Atmel

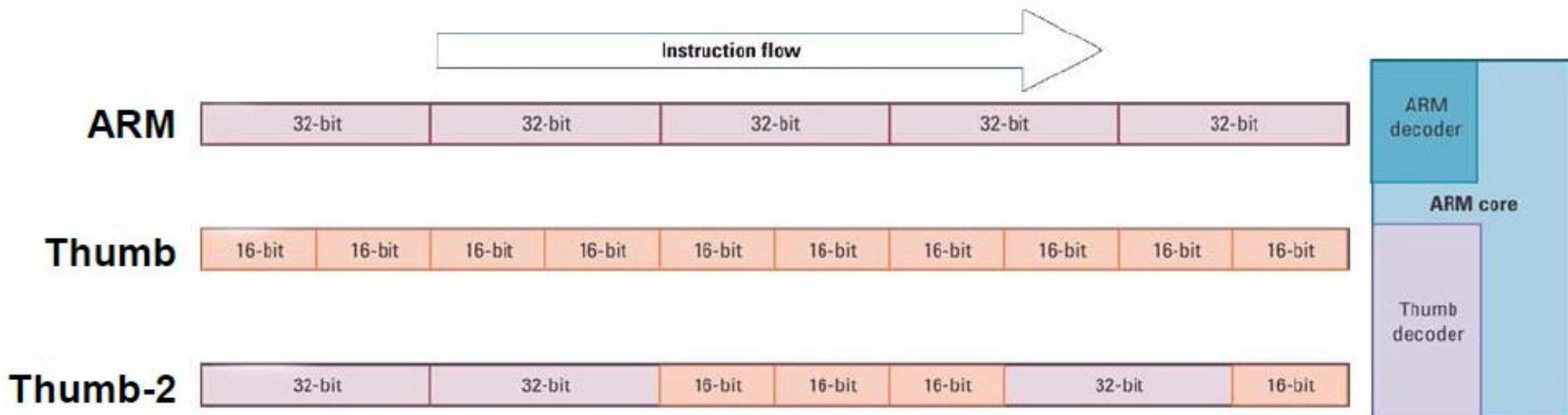
Instruction Set Architecture: ISA (2/3)

An ISA specifies,

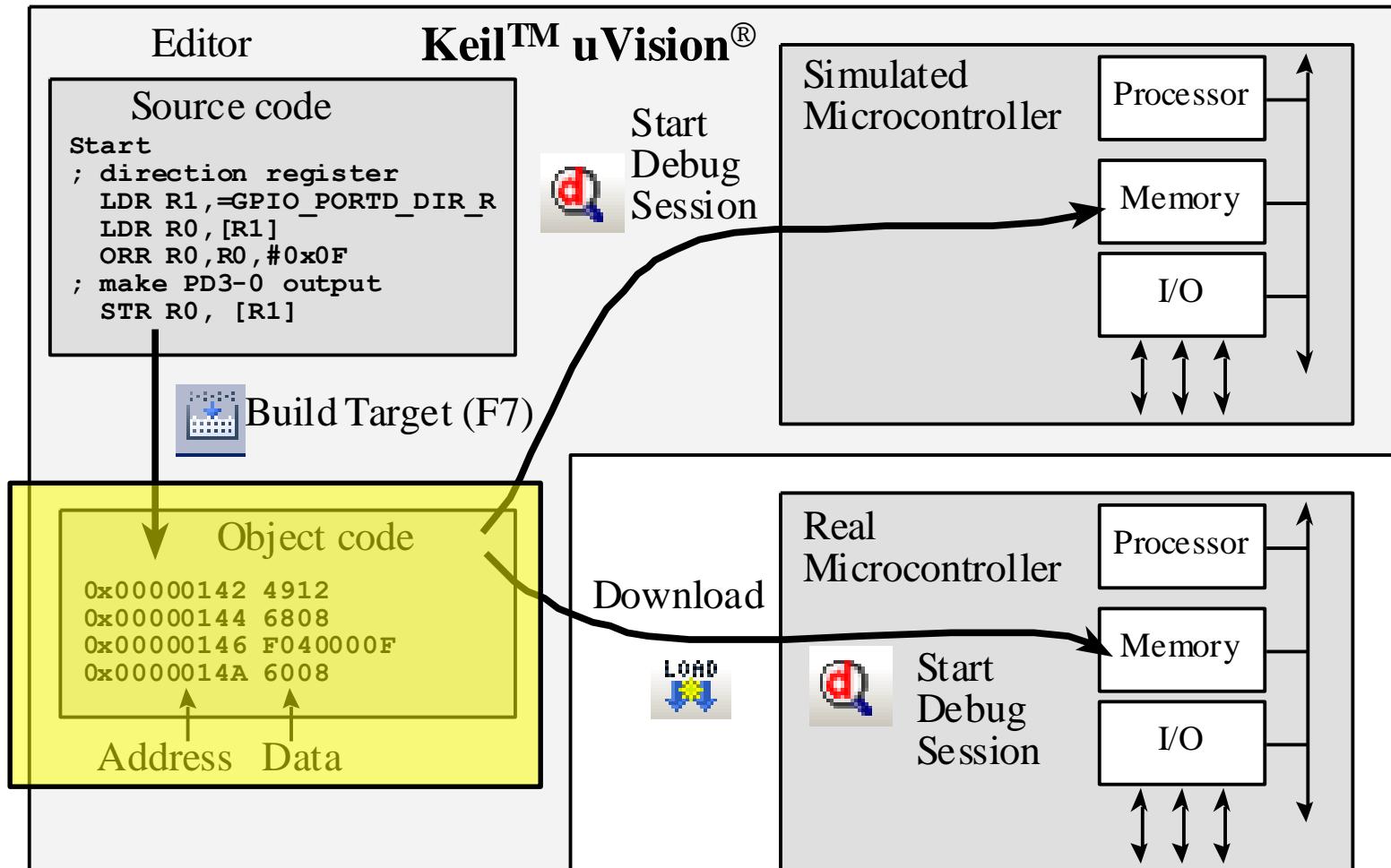
1. Supported instruction: Opcode
2. Data types (Char, Int, floating points...)
3. Registers (general purpose, special...)
4. Addressing mode for each instruction
(immediate/indexed...)
5. Memory access mechanisms for each instruction (to read from and write to memory)

Instruction Set Architecture: ISA (3/3)

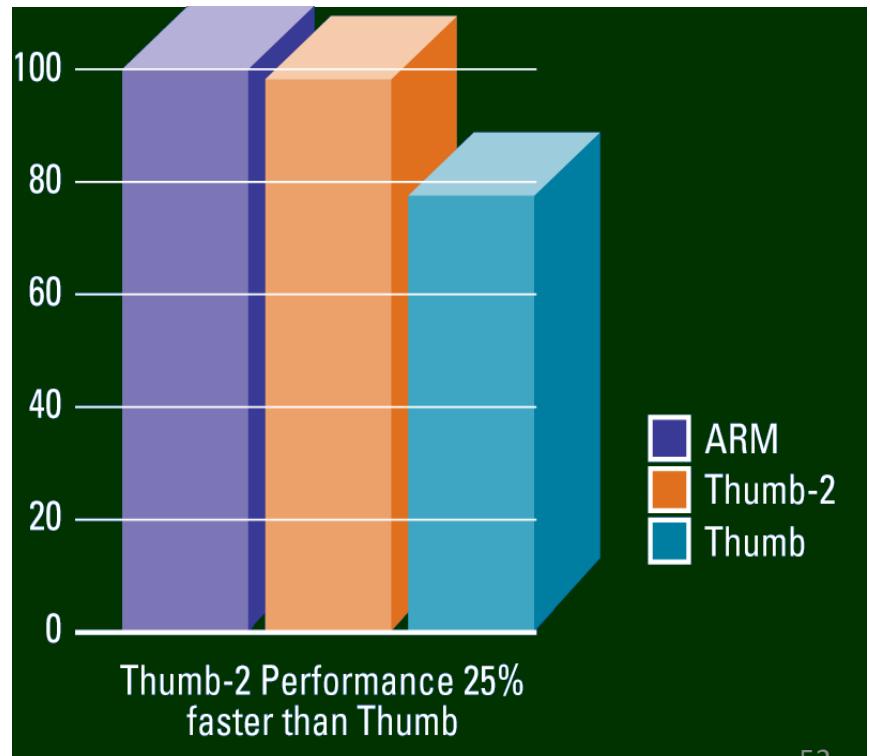
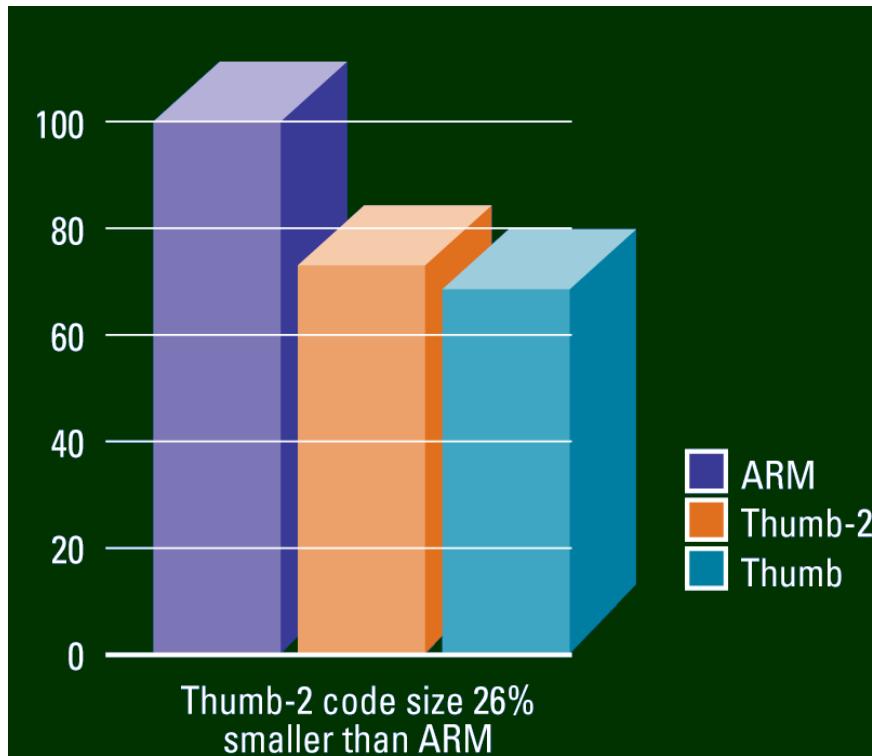
- Most ARM: 32-bit architecture
- Cortex M4 ISA: Thumb2 Instruction Set
 - Thumb-2 configuration employs both 16 and 32 bit instructions (i.e., ARM 32 + Thumb 16)
 - The Thumb-2 instruction set improves *code density* in comparison with the ARM instruction set



Details: Cortex M Architecture_32bit_devices_by_ARM.pdf



- Thumb-2 gives approximately 26% improvement in code density over ARM
- Thumb-2 gives approximately 25% improvement in performance over Thumb
- ARM Cortex M4-based family uses Thumb-2



Example: AMD Athlon (early version)



- AMD said: Athlon is compatible to the Intel x86
- AMD Athlon : A technically different but fully compatible x86 MPU
- AMD implemented the whole x86's ISA (provided by Intel) by themselves in their own way
 - Original detail design: Intel's confidential
 - ISA: Public (like a manual)
 - Thus, Windows can run on Athlon MPU but performance can be different (i.e., speed)
 - If the Windows is not working??

ARM ISA: CISC or RISC?

From : Liang Tang
To : Meiyi

ARM Instruction Set

This chapter describes the ARM instruction set.

4.1 Instruction Set Summary 4.2 The Condition Field 4.3 Branch and Exchange (BX) 4.4 Branch and Branch with Link (B, BL) 4.5 Data Processing 4.6 PSR Transfer (MRS, MSR) 4.7 Multiply and Multiply-Accumulate (MUL, MLA) 4.8 Multiply Long and Multiply-Accumulate Long (MULL, MLLA) 4.9 Single Data Transfer (LDR, STR) 4.10 Halfword and Signed Data Transfer 4.11 Block Data Transfer (LDM, STM) 4.12 Single Data Swap (SWP) 4.13 Software Interrupt (SWI) 4.14 Coprocessor Data Operations (CDP) 4.15 Coprocessor Data Transfers (LDC, STC) 4.16 Coprocessor Register Transfers (MRC, MCR) 4.17 Undefined Instruction 4.18 Instruction Set Examples	4.2 4.5 4.6 4.8 4.9 4.10 4.11 4.12 4.13 4.14 4.15 4.16 4.17 4.18
--	---

ARM Instruction Set

4.1 Instruction Set Summary

4.1.1 Format summary

The ARM instruction set formats are shown below.

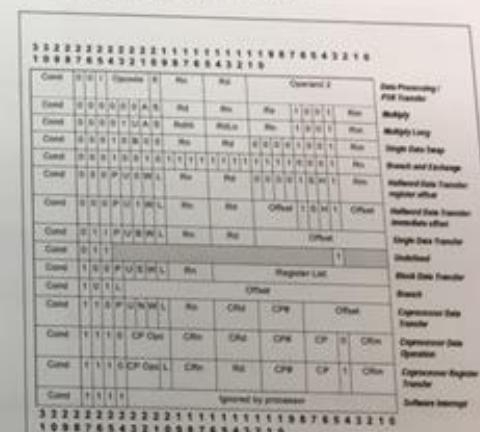


Figure 4-5 illustrates the ARM instruction set format. It shows two main formats: Format 1 and Format 2. Format 1 consists of a 3-bit Condition field (Cn), a 1-bit Operation field (Op), and three 4-bit Register fields (Rn, Rd, Rm). Format 2 consists of a 3-bit Condition field (Cn), a 1-bit Operation field (Op), and three 4-bit Register fields (Rn, Rd, Rm), followed by a 12-bit Offset field. The diagram also includes a legend for various instruction types:

- Data Processing: P/M/R/M
- Multiply: M
- Multiply Long: ML
- Single Data Swap: SWSW
- Branch and Exchange: BX
- Halfword Data Transfer: HMDR
- Halfword Data Transfer Immediate Effect: HMDRIM
- Single Data Transfer: LDR/STR
- Undefined: UNDEF
- Block Data Transfer: LDM/STM
- Block Data Header: BDH
- Branch: B
- Compressed Data Transfer: CDT
- Compressed Data Operator: CDO
- Compressed Register Transfer: CDT
- Software Interrupt: SWI

Note: Some instruction codes are not defined but do not cause the Undefined instruction to be taken, for instance a Multiply instruction with bit 6 changed to a 1. These instructions should not be used, as their action may change in future ARM implementations.

ARM7TDMI-S Data Sheet
ARM CDI 0584D

Final - Open Access

ARM7TDMI-S Data Sheet
ARM CDI 0584D

Final - Open Access

ARM ISA: CISC or RISC?

- RISC (Reduced Instruction Set Computer)
 - Few instructions
 - Instructions execute in 1 or 2 bus cycles
 - Instructions have fixed lengths: word size
 - i.e., MIPS, AVR (Atmel), PowerPC (IBM), ARM, etc.
- CISC (Complex Instruction Set Computer)
 - Many instructions
 - Instructions execute varying in time
 - Instructions have varying lengths
 - i.e., Intel X86, IBM360/370, etc.

Why RISC is used in ARM?

- Top 10 instructions for 80x86

Rank	Instruction	Average Percent total executed
1	load	22%
2	conditional branch	20%
3	compare	16%
4	store	12%
5	add	8%
6	and	6%
7	sub	5%
8	move register-register	4%
9	call	1%
10	return	1%
Total		96%

- Simple instructions dominate instruction frequency:
R means just reduced instruction size (#)

- Each instruction requires physical implementation → space, cost, logic gates
- Supporting many instructions = Expensive and large size

RISC advantage over CISC

- Low complexity: Generally results in overall speedup
- Very-large-scale integration (VLSI) implementation advantages: Less transistors
- Marketing: Reduced design time, less errors, and more options to increase competitiveness

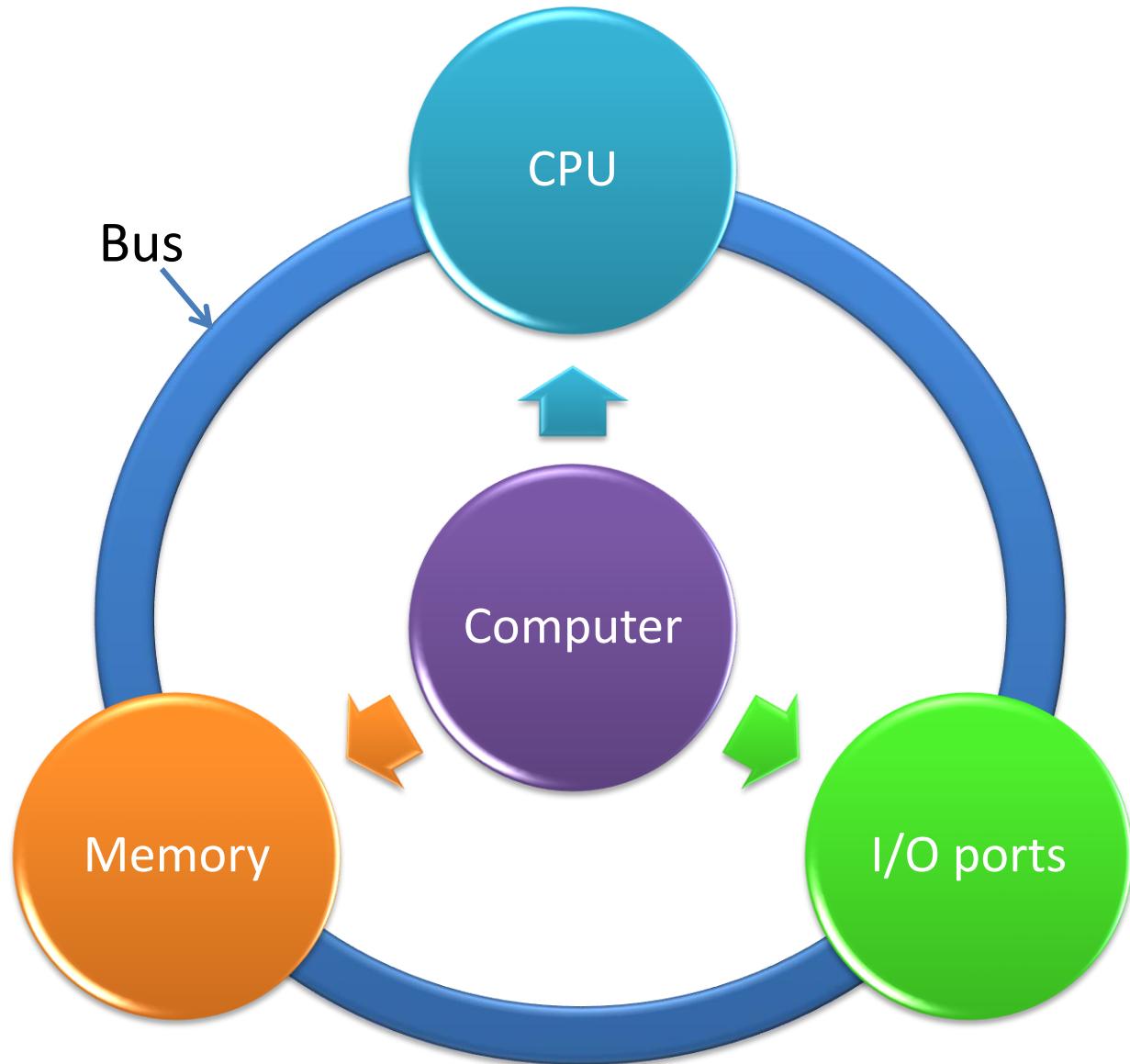
The computer architecture

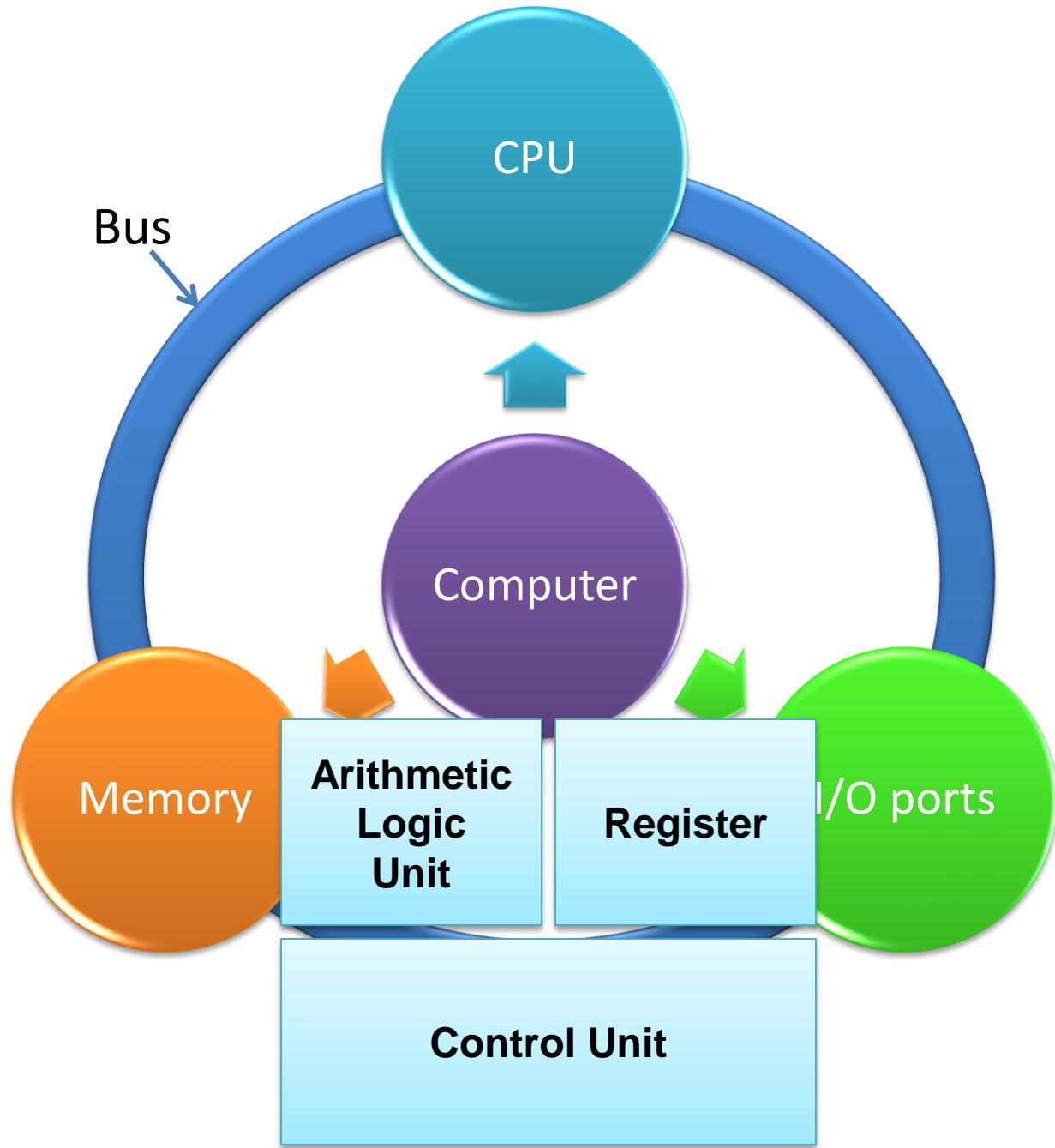
❑ Instruction Set Architecture: ISA

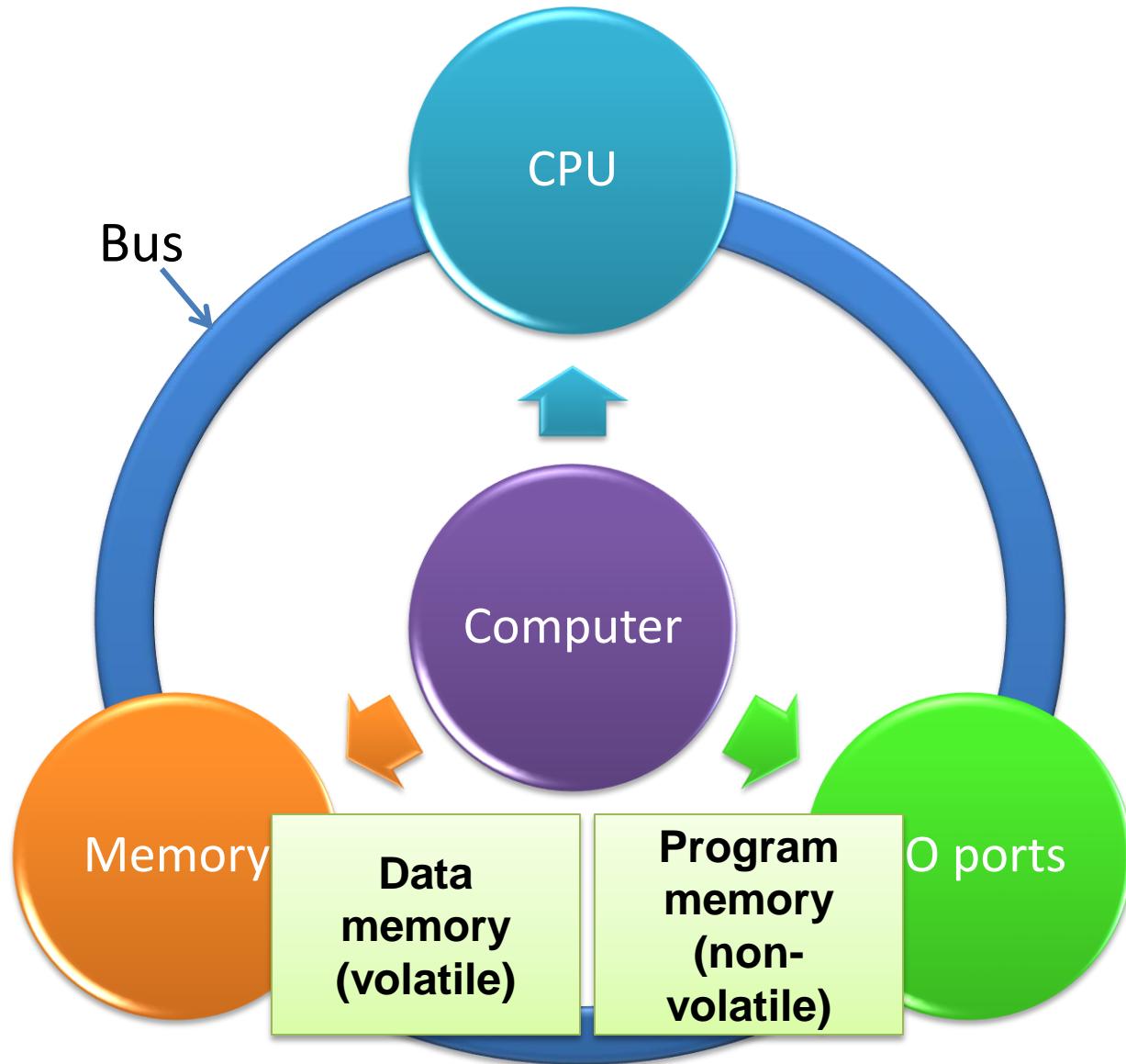
- RISC (Reduced instruction set computer)
- CISC (Complex instruction set computer)

■ Processor design

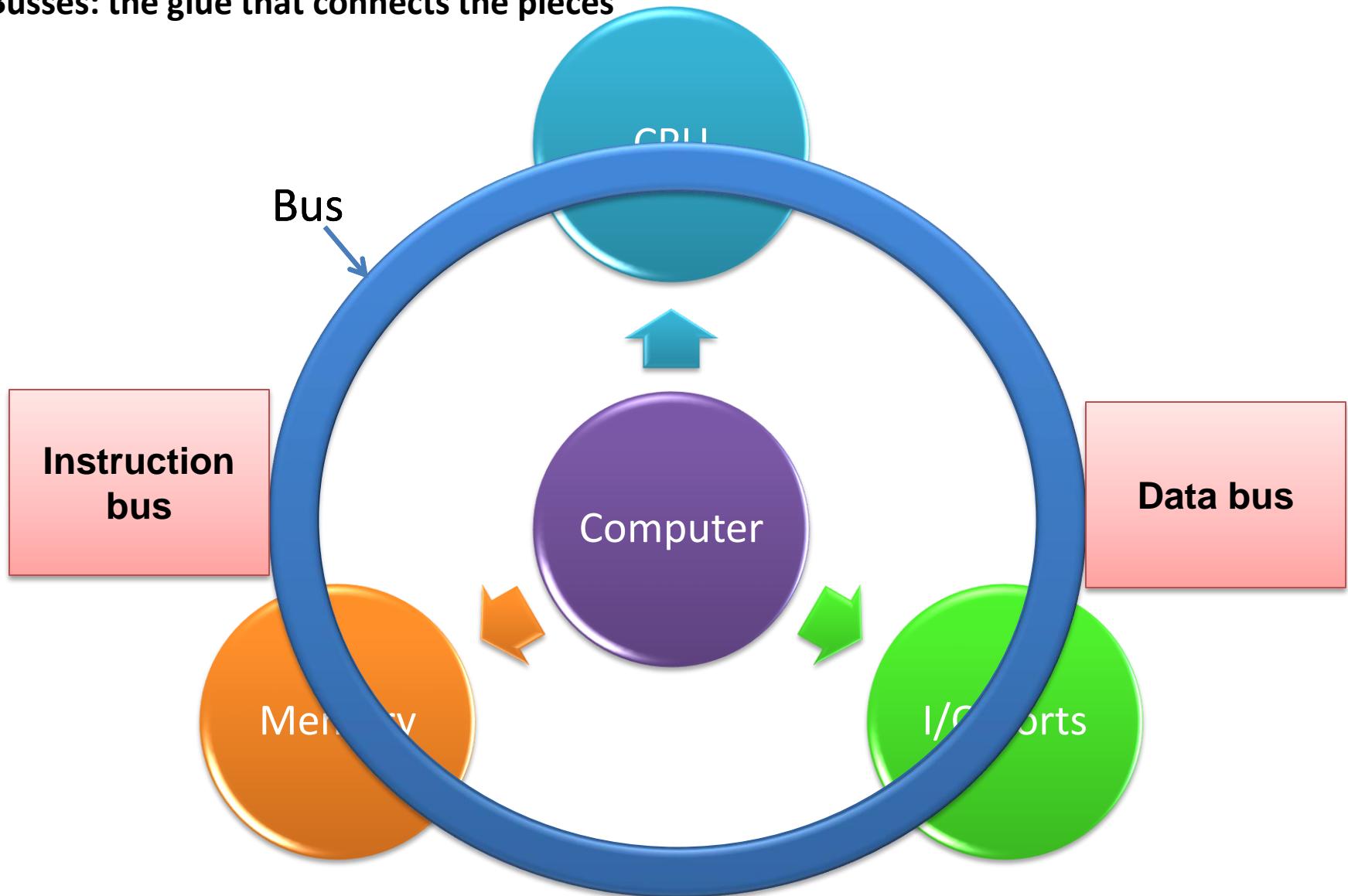
- Von Neumann architecture
- Harvard architecture







Busses: the glue that connects the pieces



- Computer parts: C_____, M_____, I/_ , B__
 - CPU or processor:
 - C_____
 - A__
 - R_____
 - Memory:
 - D___(i.e., RAM, volatile)
 - P_____ (i.e., ROM, non-volatile)
 - Bus: collection of signals (wires)
 - D___ (Read/write data from RAM or I/O, fetch opcodes from RAM),
 - I_____ (Fetch opcodes from ROM)

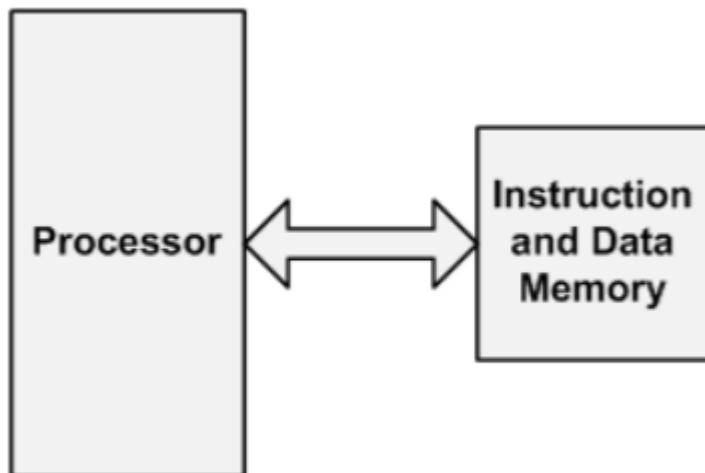
- I/O ports: physical connections between the computer and the world
 - Information enters (I ports) and exits (O ports)
 - A port: a collection of pins which can be used for either input or output
 - A pin: a place of input or output where an actual signal is connected to the microcontroller
- I/O interfaces
 - Hardware components (external to the computer, the input port) + software
 - All together perform the I/O function.

Core architecture

Von-Neumann

Instructions and data are stored in the same memory

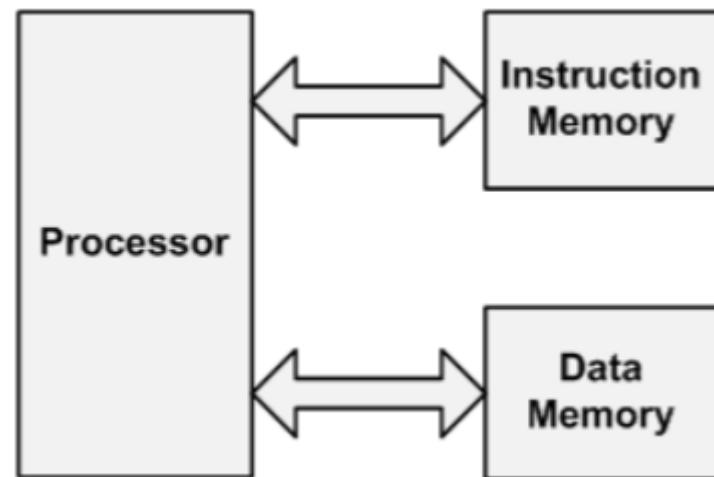
- Simple and inexpensive
- Access to data or instruction, one at a time



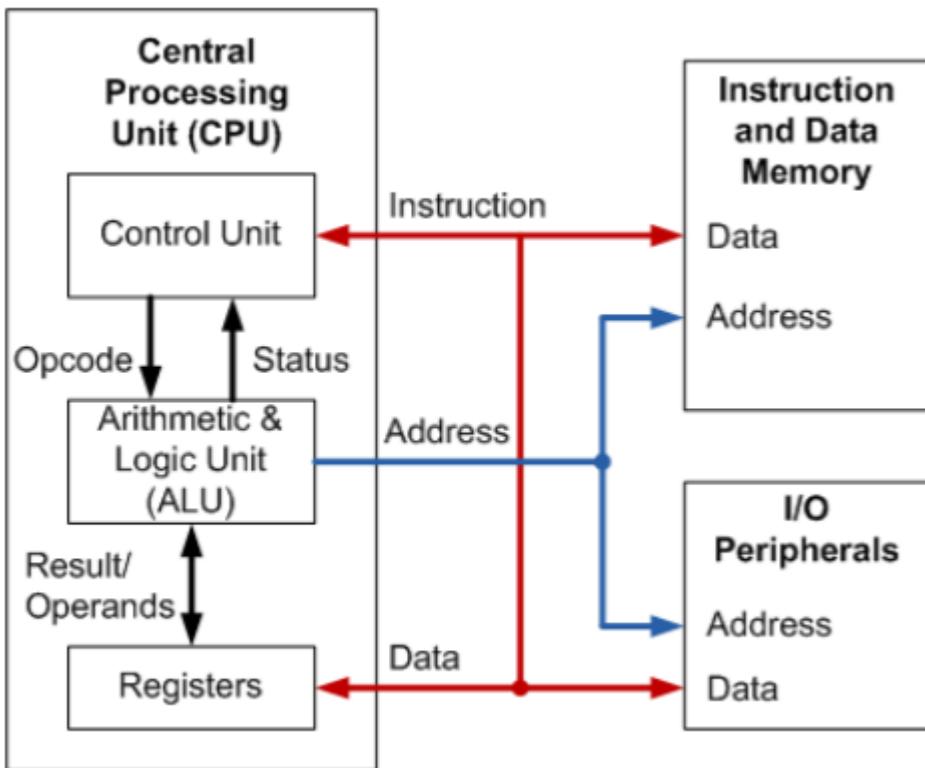
Harvard

Data and instructions are stored into separate memories

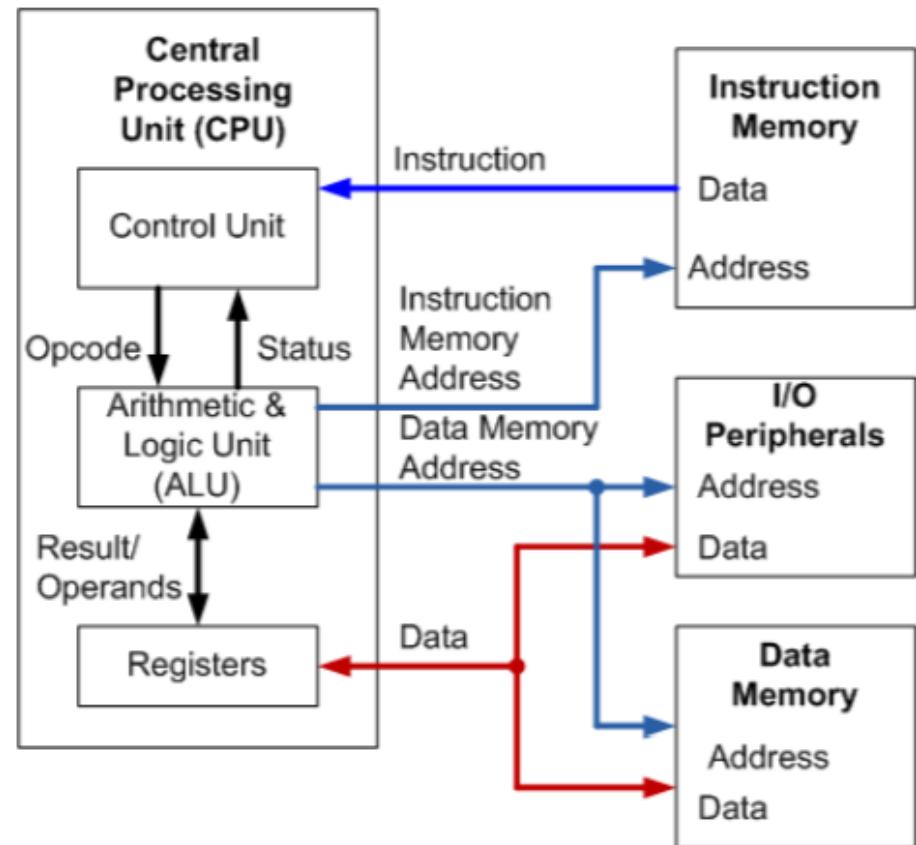
- Faster
- More energy efficient
- Different bus sizes



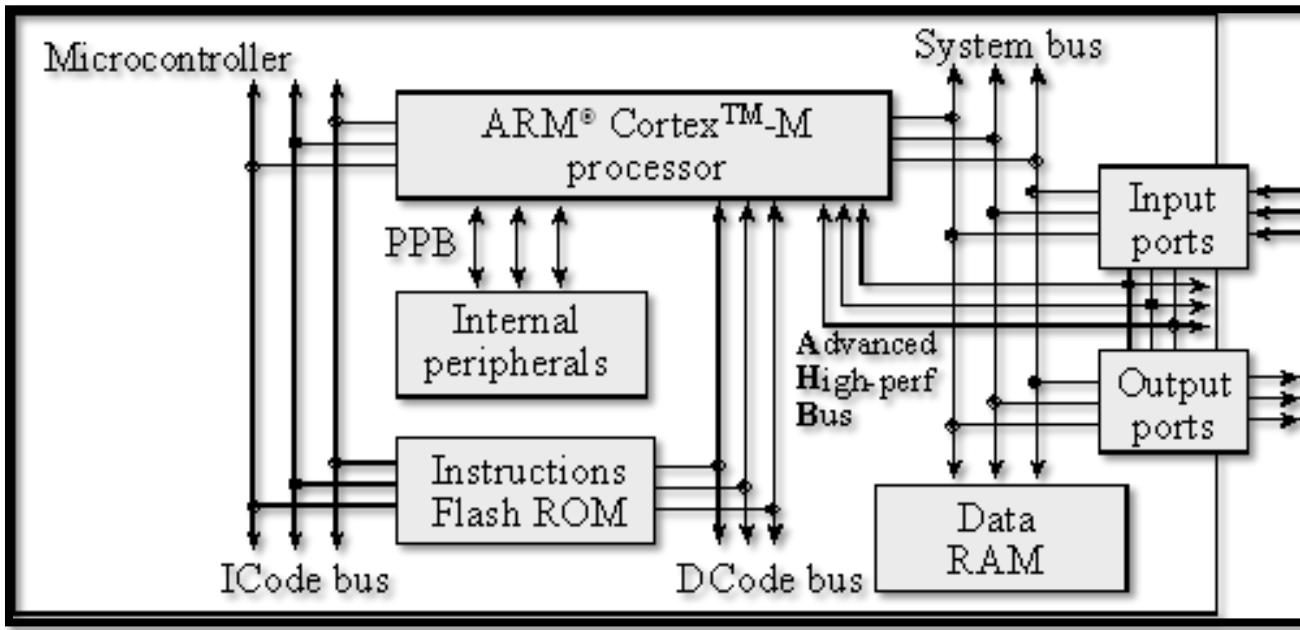
Von-Neumann



Harvard



ARM® Cortex-M: Harvard architecture



Harvard architecture of an ARM® Cortex-M-based microcontroller

- Instructions and data are separated

AMBA?

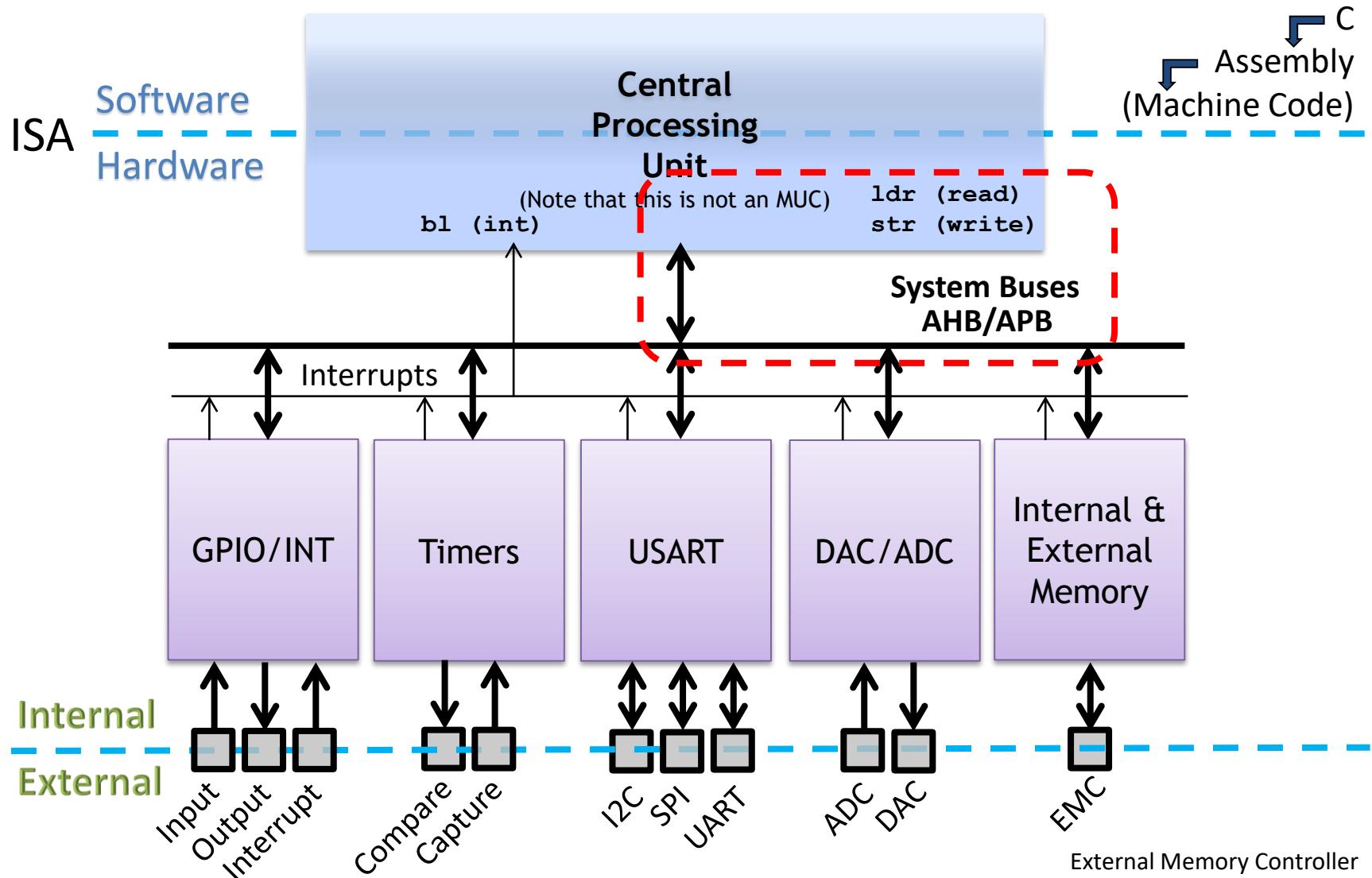
- Advanced Microcontroller Bus **Architecture**
- From Wikipedia, the free encyclopedia
 - The AMBA is an open-standard, on-chip interconnect specification for the connection and management of functional blocks in system-on-a-chip (SoC) designs.

Contents [hide]

- 1 Design principles
- 2 AMBA protocol specifications
 - 2.1 AXI Coherency Extensions (ACE and ACE-Lite)
 - 2.2 Advanced eXtensible Interface (AXI)
 - 2.3 Advanced High-performance Bus (AHB)
 - 2.4 Advanced Peripheral Bus (APB)
- 3 AMBA products
- 4 Competitors
- 5 See also
- 6 References
- 7 External links

https://en.wikipedia.org/wiki/Advanced_Microcontroller_Bus_Architecture

Busses: the glue that connects the pieces

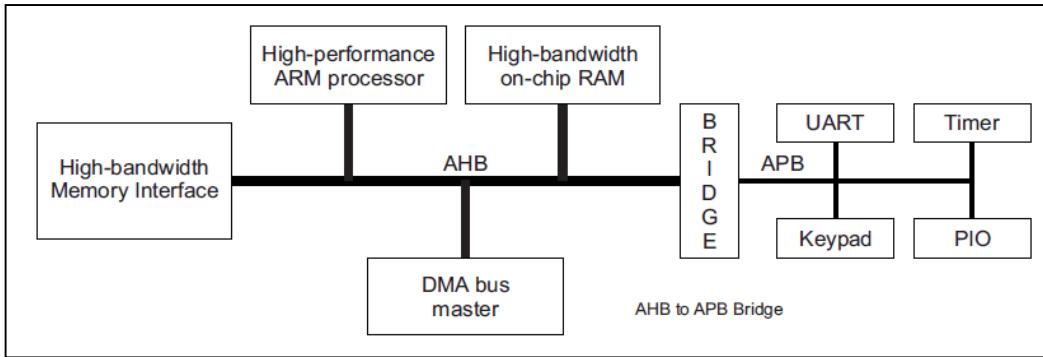


→ Advanced/AMBA High-performance Bus (AHB)

→ Advanced Peripheral Bus (APB)

Page 48 of tm4c123 datasheet

Figure 1-1. Tiva™ TM4C123GH6PM Microcontroller High-Level Block Diagram



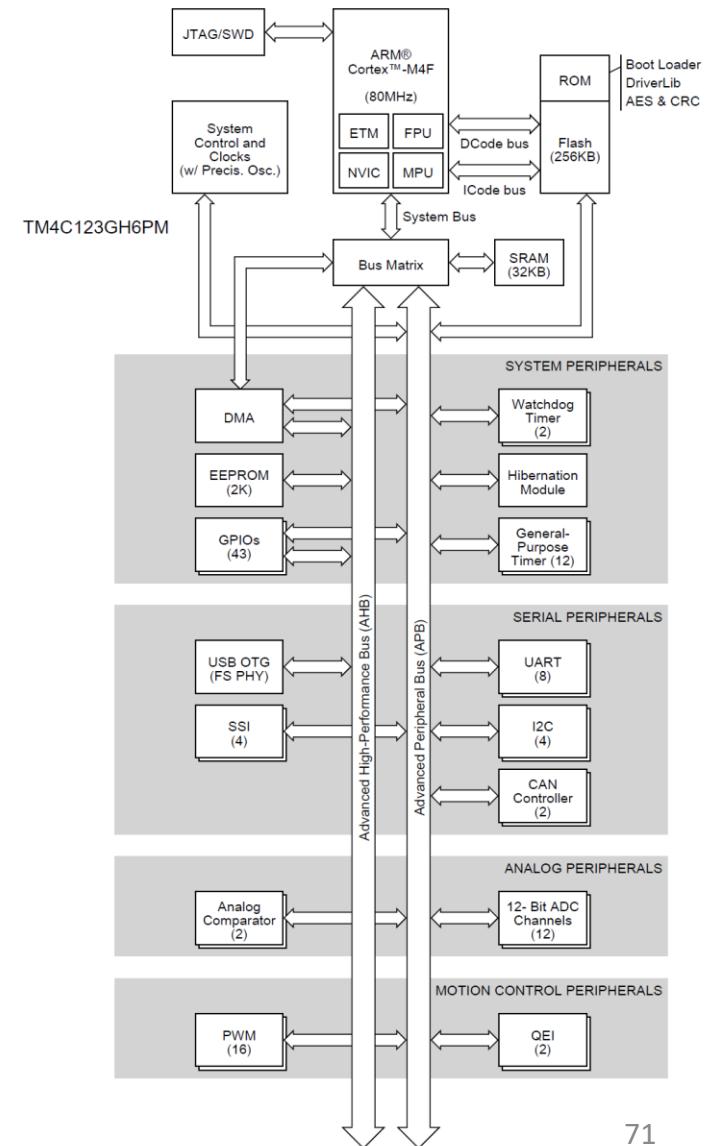
AHB/APB General structure

AHB

- High performance
- Pipelined operation
- Burst transfers
- Multiple bus masters
- Split transactions

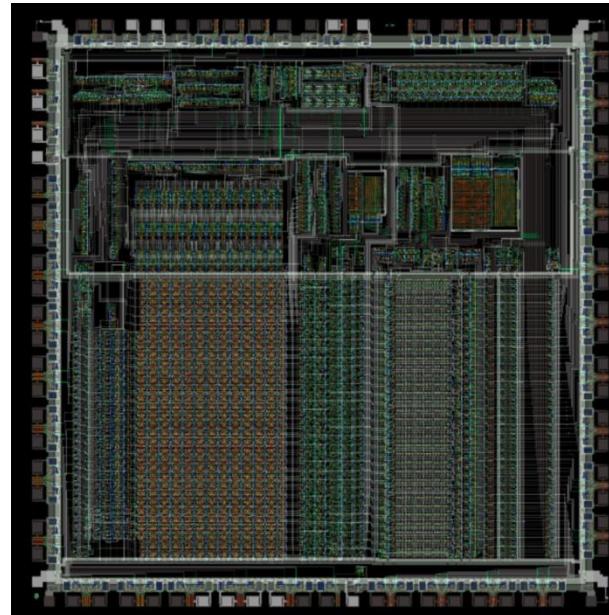
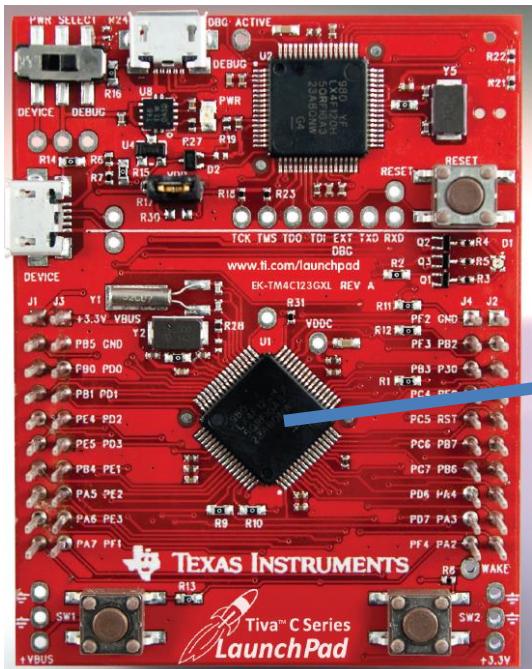
APB

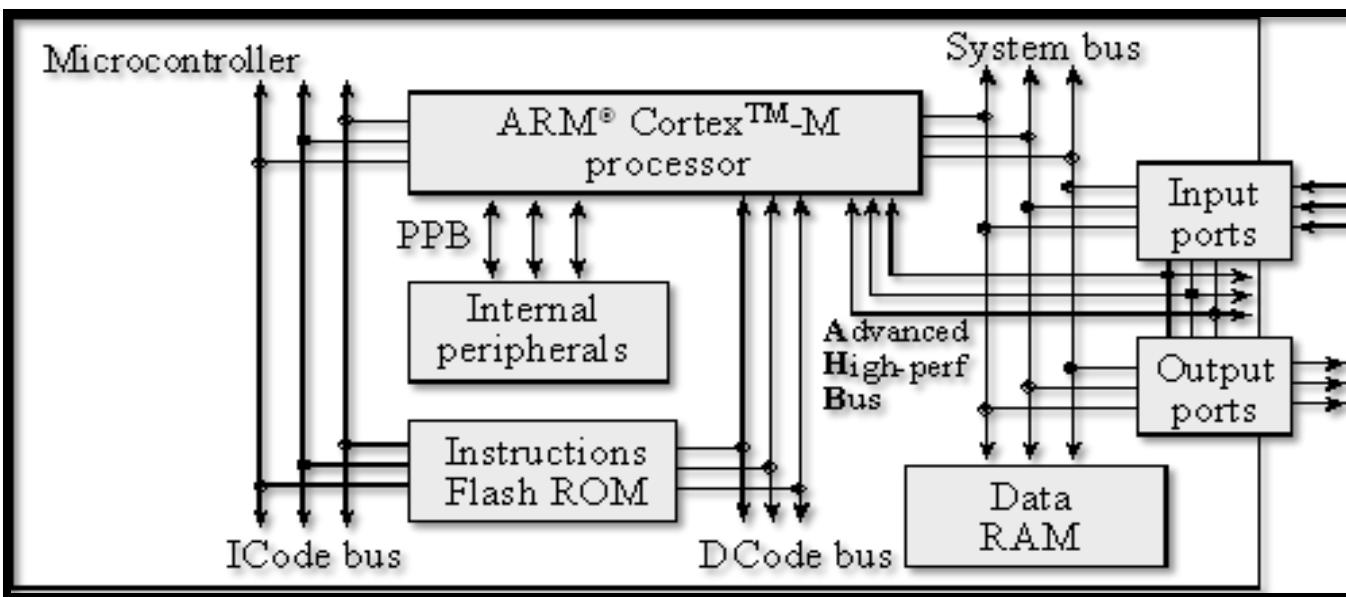
- Low power
- Latched address/control
- Simple interface
- Suitable of many peripherals



The MCU chip on our launch board

- An ARM® Cortex-M-based MCU: TM4C123GH6PMI
- Harvard architecture





Reuse image: Harvard architecture of an ARM® Cortex-M-based microcontroller

- Data and instruction fetching buses are separated
- Five buses
 - ICode bus: fetch opcodes from ROM
 - Dcode bus: read constant data from ROM
 - System bus: R/W data from RAM or I/O, fetch opcode from RAM
 - PPB: R/W data from internal peripherals like NVIC
 - AHB: R/W data from high-speed I/O and parallel ports (M4 only)

Summary of your own

Reading

Vol.1	Vol.2
Ch.1 (1.7)	Ch.1 (1.1, 1.2)
Ch.2 (2.1, 2.2...)	Ch.2 (2.1,
Ch.3 (3.1, 3.5)	2.2, 2.7...)
Ch.4 (4.1)	